Universitat Autònoma de Barcelona

Writer Identification by a Combination of Graphical Features in the Framework of Old Handwritten Music Scores

A dissertation submitted by **Alicia Fornés Bisquerra** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctora en Informàtica**.

Bellaterra, May 2009

Director: **Dr. Josep Lladós Canet**
Universitat Autònoma de Barcelona
Dep. Ciències de la Computació & Centre de Visió per Computador
Co-director: **Dra. Gemma Sánchez Albaladejo**
Universitat Autònoma de Barcelona
Dep. Ciències de la Computació & Centre de Visió per Computador

Centre de Visió
per Computador

This document was typeset by the author using LaTeX 2_ε.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN: 978-84-936529-9-9

Printed by Ediciones Gráficas Rey, S.L.

This thesis is dedicated to my family

Agraïments

Una tesi és un plat que es fa poquet a poquet, cuinat a foc lent, amb constància i paciència... i més constància i paciència... (***while*** *paciència* > 0 ***do*** *... tesi!!!* ***end***)... que a vegades penses que no s'acaba mai (ostres!!, s'ha tornat un bucle infinit??) ... però arriba un dia que s'acaba (increïble, però cert!), i te n'adones que ja han passat gairebé 6 anys des que vas venir de Mallorca per fer un doctorat. El temps passa molt, molt aviat, han passat moltes coses, i has conegut moltes persones, que de diferents maneres t'han ajudat molt, i que es mereixen ser agraïdes en aquestes línies.

Primer de tot voldria agrair als meus directors de tesi Josep Lladós i Gemma Sánchez, pels seus consells, dedicació i suport durant tots aquests anys. En especial al Josep, que tot i tenir una muntanya de feina pels diferents càrrecs i responsabilitats, sempre treia temps de sota les pedres per revisar-me la feina (el temps és infinit m'has dit alguna vegada, es pot allargar tant com es vulgui, com el DTW). A més, voldria agrair la seva actitud propera, suportant amb paciència les meves *xerrades*.

En segon lloc voldria agrair al Juan José Villanueva, al Centre de Visió per Computador i a la Universitat Autònoma de Barcelona per donar-me l'oportunitat de realitzar aquest treball de recerca, així com a l'AGAUR per concedir-me la beca per a l'estada de recerca a Suïssa. També voldria agrair al Josep Maria Gregori i Joan Casals del departament d'Art de la UAB, per la seva ajuda en l'accés a les partitures antigues del Seminari de Barcelona, Terrassa i Canet de Mar.

The two research stays in Bern have been very important to me. I would like to thank Prof. Dr. Horst Bunke for his great supervision, guidance and advice during these five months. It has been a pleasure for me. I would also like to thank Andreas Schlapbach and Vivian Kilchherr, for providing support in the experiments; and also to the FKI people, Roman Bertolami, Volkmar Frinken, Kaspar Riesen, Andreas Fischer, Marcus Liwicki and Emanuel Indermuhle, for making me feel as a member of the group. Special thanks to Susanne Thüler, for letting me stay in her house, and also for looking after me, as her daughter.

Voldria agrair també a la gent del DAG pel suport, idees i brainstormings: Agnès, Joan, Marçal, Jose, Miquel, Ernest, Dimos, Mathieu, Partha, Jaume, Albert, Farshad, Antonio, Henry, Ricard i Enric. Gràcies a tota la gent del CVC, que fan del centre un lloc de treball agradable. Agrair al personal d'informàtica i administració, que fan que el CVC funcioni com cal: Montse, Pilar, Joan Masoliver, Raquel, Ana, Helena, Mari... Menció especial mereixen els companys de despatx, de dinars, cafes i descansos: Joan, Marçal, Eduard, Xavi, Aura, Agnès, Àgata, Xevi, Carles, Javi, Ferran, Ignasi, Dani, Jaume, David... gràcies pels bons moments que he passat amb vosaltres.

També voldria donar gràcies als que han col.laborat de moltes diferents maneres en aquest treball de recerca: Ricard, Henry, Dimos, Xavi Otazu, Joan, Petia, Oriol... així com als voluntaris que desinteressadament han ajudat a crear la base de dades de símbols musicals. Querría destacar a Sergio, mi *blurred-colega*, compañero de cafés, paseos y *filosofadas* varias que suelen acabar en publicaciones, gracias por despegarme (literalmente) de la silla del ordenador y obligarme a estirar las piernas!

Donat que la tesi es fa després d'una carrera universitària, no voldria descuidar un agraïment als professors d'informàtica la Universitat de les Illes Balears, la Universitat de València i la Universitat Autònoma de Barcelona. Especialment a l'Albert Llamosí, pels seus consells durant i després de la carrera.

En una tesi informàtico-musical, no pot mancar un agraïment especial a tots els professors de música que han passat per la meva vida. Tots i cada un d'ells m'han contagiat la passió per la música, tant la clàssica com la moderna. També voldria agrair als melòmans Xavi, Eli i Marçal, que m'ajuden a mantenir viva la flama de la passió per la música, convidant-me a descobrir nous compositors cada dia. Recordo encara les paraules que fa molt temps pronunciava en Santiago Francia, que deia que sense música la vida es veuria en blanc i negre. La música és com una teràpia, i confesso que sóc melòmana-dependent gràcies a tots vosaltres.

Voldria agrair als amics, que suporten tant els bons com els mals moments. Primerament, agrair als amics mallorquins: Juanan, Javi, Sven, Noemí, Sebas, Cris, Víctor, Camila, Gabriela, Gabri... per estar sempre disponibles, amb l'amistat que dura i dura a pesar de la distància, que em recorden les meves arrels, així com de lo molt enamorats que estem els mallorquins de la mar. També voldria agrair a la gent dels Caputxins, Oblates i voluntaris, per fer-me obrir els ulls al món, mostrant-me que a la vida hi ha alguna cosa més que passar-s'ho bé. I òbviament, agrair també als nous amics i companys de pis trobats a Barcelona, que eviten que m'enyori: Mònica, Virginia, Bàrbara, Ester, Anna, Mar, Yolanda, Noe, Edu, Juli, Anna, Mónica, Dani i *Maxitus*... Agradecer especialmente a Rosa y Eli, por acogerme y cuidarme como a una más de la familia... y por la de veces que me he ahorrado tener que cocinar :-)

Voldria acabar donant les gràcies a la meva família, per l'estima i suport durant tots aquests anys, per sentir-vos propers, tot i la distància. Sense vosaltres puc assegurar que no hauria arribat fins aquí. Agrair a *sa* meva *germanona*, per ésser-hi sempre, sempre, per ser la meva gran confident i companya musical. A *mun* pare, no només per inculcar-me valors, sinó també pel recolzament durant tots aquests anys d'estudi (sí papà, després de tants anys... per fi he acabat els estudis!!); i a *mu* mare, que admiro molt, per estimar-me tant i per ser un exemple de donar-se als altres.

Y finalmente gracias a ti, por estar ahí, por ser mi punto de apoyo, porque como bien dice *Rosana*: *si tu no estas ahí... no se...*

Resum

L'anàlisi i reconeixement d'imatges de documents històrics ha guanyat interès durant els darrers anys. La digitalització massiva juntament amb la interpretació dels documents digitalitzats permeten la preservació, l'accés i la indexació del llegat artístic, cultural i tècnic. L'anàlisi dels documents manuscrits és una subàrea d'interés excepcional. El principal interès consisteix no només en la transcripció del document a un format estàndard, sinó també en la identificació de l'autor d'un document davant d'un conjunt d'escriptors (l'anomenada identificació de l'escriptor).

La identificació de l'escriptor en documents manuscrits de text és una àrea activa d'estudi, i a la literatura és prolífica en contribucions significatives. No obstant, la identificació de l'escriptor en documents gràfics és encara un repte. El principal objectiu d'aquesta tesi és la identificació de l'escriptor de partitures musicals antigues, com a exemple de document gràfics. En referència a les partitures antigues, molts arxius històrics contenen un enorme volum de partitures musicals sense informació del seu compositor, i la recerca en aquest camp podria ser beneficiosa pels musicòlegs.

El marc de treball per a la identificació de l'escriptor proposat en aquesta tesi combina tres diferents aproximacions, corresponents a les principals contribucions científiques. La primera es basa en mètodes de reconeixement de símbols. Per a aquesta tasca, s'han proposat dos nous mètodes de reconeixement de símbols per fer front a les distorsions típiques del símbols dibuixats a mà. El primer mètode està basat en l'alineament de seqüències (Dynamic Time Warping - DTW), on els símbols es descriuen emprant seqüències de vectors, i la proximitat entre símbols es mesura a partir de l'algoritme DTW. El segon mètode és el Model de Forma Difusa (Blurred Shape Model - BSM), que descriu els símbols emprant una funció de densitat de probabilitat que codifica la probabilitat de les densitats de les regions de la imatge.

La segona aproximació per la identificació de l'escriptor preprocéssa la imatge per obtenir línies de símbols musicals, i extreu informació de la inclinació i gruix de l'escriptura, les regions connexes, contorns i fractals. Finalment, la tercera aproximació extreu informació global, generant textures musicals a partir de les partitures, i extraient característiques de textura (filtres de Gabor i matrius de co-incidència).

Els bons resultats obtinguts demostren la idoneïtat de l'arquitectura proposada. Fins a on arriba el nostre coneixement, aquest treball és la primera contribució en la identificació de l'escriptor a partir d'imatges que contenen llenguatges gràfics.

Abstract

The analysis and recognition of historical document images has attracted growing interest in the last years. Mass digitization and document image understanding allows the preservation, access and indexation of this artistic, cultural and technical heritage. The analysis of handwritten documents is an outstanding subfield. The main interest is not only the transcription of the document to a standard format, but also, the identification of the author of a document from a set of writers (namely writer identification).

Writer identification in handwritten text documents is an active area of study, and the literature is prolific in noteworthy contributions. However, the identification of the writer of graphical documents is still a challenge. The main objective of this thesis is the identification of the writer in old music scores, as an example of graphic documents. Concerning old music scores, many historical archives contain a huge number of sheets of musical compositions without information about the composer, and the research on this field could be helpful for musicologists.

The writer identification framework proposed in this thesis combines three different writer identification approaches, which are the main scientific contributions. The first one is based on symbol recognition methods. For this purpose, two novel symbol recognition methods are proposed for coping with the typical distortions in handdrawn symbols. The first one is a Dynamic Time Warping (DTW) based method, in which symbols are described by vector sequences, and a variation of the DTW-distance is used for computing the matching distance. The second one is called the Blurred Shape Model (BSM), in which a symbol is described by a probability density function that encodes the probability of pixel densities of image regions. The second writer identification approach preprocesses the music score for obtaining music lines, and extracts information about the slant, width of the writing, connected components, contours and fractals. Then, a k-NN classifier is the used to categorize the document image. Finally, the third approach extracts global information about the writing, by generating texture images from the music scores and extracting textural features (Gabor features and co-occurrence matrices).

The high identification rates obtained in the experimental results demonstrate the suitability of the proposed ensemble architecture for the identification of the writer in music scores. To the best of our knowledge, this work is the first contribution on writer identification from images containing graphical languages.

Keywords: *Writer Identification, Graphics Recognition, Symbol Recognition, Optical Music Recognition.*

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This Chapter presents the motivation and objectives of the thesis. We briefly overview the Document Analysis and Recognition research field, including not only the analysis of historical documents but also the analysis of graphic documents and their main associated concepts. Afterwards, we introduce the writer identification problem, and discuss about its applicability to graphic documents, such as music scores. We also discuss about the discriminant graphical properties used by musicologists for writer identification. Finally, we overview the main difficulties found in such a task, and summarize the objectives and contribution of this work.

1.1 Motivation

Document Image Analysis and Recognition (DIAR) is an important field in Pattern Recognition, whose aim is the analysis of contents of document images. It has three main research directions: text recognition, graphics recognition and layout analysis. Document analysis in handwritten historical documents has attracted growing interest in the last years, whose aim is the conversion of these documents into digital libraries, helping in the diffusion and preservation of artistic and cultural heritage. In addition to the preservation in digital format, the interest of aplying DIAR to historical handwritten documents is twofold. The first is the recognition and transcription of the document to a machine readable format, while the second consists in the classification of the document, such as the identification of the authorship of the document (namely, writer identification).

Writer identification consists in determining the author of a piece of handwriting among a set of writers. It is an important task for the automatic processing of documents, allowing applications such as forensic document examination, in which the handwriting can be used for identification (such as the signature verification in bank checks, or the recognition of the voice, face, iris and fingerprints), and the analysis of digital libraries (e.g. classification of documents, retrieval by content).

Writer identification in handwritten text documents has been an active area of study since many years (see [STB00], [SB08], [SB04]), whereas the identification of

the writer of graphical documents is still a challenge. Graphic documents make use of graphical languages (composed by symbols and combination rules) for describing ideas in a compact way. Referring handwritten ones, writer identification can be performed analyzing the symbols appearing in these documents, because it has been shown that the author's handwriting style that characterizes a piece of text is also present in a graphic document.

Music scores are an example of hybrid documents (because they contain both graphics and text) with an important research community. A growing interest has been the analysis of ancient music scores (see [Car95], [PVS03]), with the purpose of the preservation of cultural heritage. In fact, after the digitization of historical documents, an important application is the retrieval of anonymous documents for their analysis, and the validation of the authorship of some documents. Since many historical archives contain a huge number of sheets of musical compositions without information about the composer, musicologists must work hard for identifying the writer of every sheet. As far as we know, only one project (*eNoteHistory* [1] [BIM04], [GÖ3]) has been performed about writer identification in music scores. However, no quantitative results have been published, and as far as we know, this work has not been continued. For that reason, a writer identification approach for old music scores is still required for helping musicologists in such a task, which is time consuming and prone to errors. In this context, the handwriting style of the hand-drawn music symbols can be used for determining the authorship of a music score.

As a summary, there is a great interest in identifying the authorship of graphical documents, and the main motivation of this thesis is to perform writer identification in old music scores, as an example of graphical documents.

1.2 Context: Recognition and Identification of Historical Graphical Documents

In this section a brief review of historical document analysis and graphical documents is performed. First, an overview of historical documents and digital libraries is performed. Secondly, graphic documents are presented, including an introduction to the structure and notation of music scores. Finally, the characteristic properties used for identifying the writer of a music score are discussed.

1.2.1 Historical Document Analysis and Digital Libraries

Document analysis in historical documents has attracted growing interest in the last years, whose aim is the conversion of documents into digital libraries. It is an unquestionable fact that the knowledge contained in books and paper documents carries a great historical, cultural, scientific and social value, and the research in historical documents will help in the diffusion, accessibility and preservation of cultural heritage. In fact, there are some difficulties for accessing to historic documents. Although there is a huge amount of old documents in Archives and Churches all over Europe, the access is allowed only to some expert historians, because of safety reasons (documents

[1] http://www.enotehistory.de

are very valuable), and also because of the delicate state of the paper (there is an important degree of paper degradation).

History of Digital Libraries (DL) of documents starts in the sixties when the evolution of computer science allows considering digitization as a way to provide a better and wider access to paper archives and preserve them from time degradation [Les97]. Since then, many projects have been undertaken by many libraries and other organizations worldwide, such as the *Project Gutenberg* (1971), the *Project Perseus* (1987) and *iBiblio* (1992). Nowadays, it is a common practice among institutions to create DLs for storing, organizing and accessing large collections of documents. Therefore, their use has been widely spread (see [BJFD98], [DL96],[MCF00]) and some big projects have been undertaken, such as *Google Books* [2], the *Million Book Project* [3] or the *European Library* [4], the european research projects *Debora* [5], *Impact* [6], *Europeana* [7]; or the national research projects *iDoc* [8], *NaviDoMass* [9], among others.

Digital content infrastructures will greatly benefit from a digital representation of the knowledge enclosed in document collections. After the digitization of paper documents, the extraction of information from the document image is required. It must be said that historical documents are difficult to process automatically due to paper degradation, the show-through and bleed-through effect, and the common lack of a standard notation. In addition, the presence of handwritten text, graphical illustrations or both in historical documents is common (see an example in Fig.1.1).

Figure 1.1: Examples of old documents.

DIAR covers three main research areas: text recognition, layout understanding and graphics recognition. In some recent papers reviewing the state-of-the-art of document analysis techniques for DLs (see [Bai04],[SAP+06]), some of these challenges are identified, which are mainly the following. There are some problems with the

[2] http://books.google.com
[3] http://www.ulib.org
[4] http://www.theeuropeanlibrary.org/
[5] http://debora.enssib.fr/
[6] http://www.impact-project.eu/
[7] http://www.europeana.eu/
[8] http://web.iti.upv.es/ prhlt/content.php?page=projects/handwritten/idoc/idoc.php
[9] http://l3iexp.univ-lr.fr/navidomass

image capture and digitization, because the degradation of the paper document is usually important, and also, document enhancement techniques are required for deskewing, de-warping, binarizing and removing pepper noise, among others. There is also a need to improve the existing Document Image Analysis methods for analyzing the whole document content, as well as methods for improving the presentation, display, indexing and retrieval.

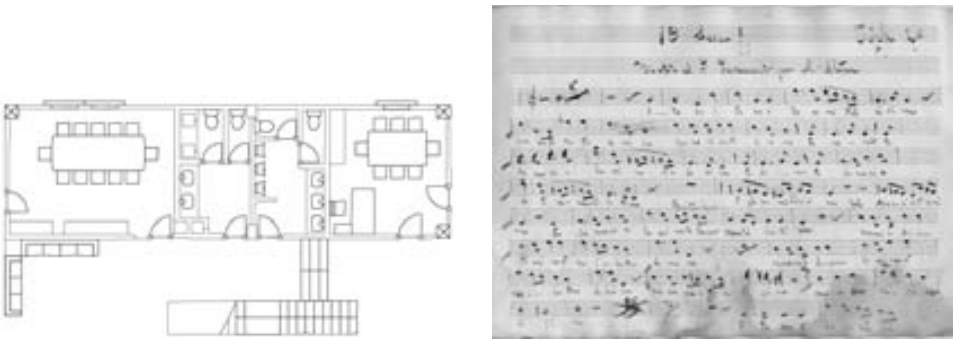
Recent works include methods that deal with different aspects of the recognition, such as the layout analysis in documents from the Archive of the Cabinet of the Dutch Queen [BvKS+07], a language to describe tables in damaged handwritten documents from the 19th century [MCC08], a description system for military forms of the 19th century [Coü06], a segmentation approach for old color maps [RBO08], word segmentation in degraded documents [MNG07], an OCR for old documents [VGSP08], an approach for recognizing broken characters [LSS07], or a word spotting technique for indexing historical documents [LS07]. We can also remark an approach for the reconstruction of Don Quixote [Spi04], a recognition system for byzantine chants [DMP08], the characterization of pictures of old docs based on a texture approach [JME+07], ancient ornamental letter indexing [KUKO08] and retrieval [DJJ08], the recognition of korean [KCKK04] and greek [GNP+04] handwritten documents, among others.

1.2.2 Overview of Graphic Document Recognition

In the field of Pattern Recognition and Document Analysis, the recognition of graphical documents has been an area of intensive research, which has been applied to a large number of domains like engineering, architecture, software modelling, music, cartography, etc. [LVSM02]. Some examples of graphic documents can be seen in Fig.1.2. Each kind of graphic-rich document has associated its specific graphical language which convey important information.

Graphical languages are expressive and synthetic tools for communicating ideas in some domains, and allow users to describe complex models with compact diagrammatic notations. A graphical language consists of an alphabet of symbols (defined as synthetic visual entities) and rules or productions referring to the relationships between the symbols. Thanks to the recognition of the alphabet of symbols of these graphical languages and their relations, combined with domain-dependent knowledge, the whole document has a meaning, allowing its automatic processing.

Concerning the Symbol Recognition research field, a growing interest in the last years has been the recognition of hand-drawn symbols appearing in graphical documents. Some techniques used for shape recognition have been applied to symbol recognition, and also, some specific symbol recognition methods have been proposed. There are several wide areas of application of hand drawn symbol recognition: one corresponds to sketching frameworks, in which the communication between users and computers is achieved through free hand drawings or gestures (a gesture is a set of strokes with an associated command); another one includes existing textual manuscripts, specifically collections of scanned documents in libraries and archives rich in graphical information where the recognition of symbols can be useful for transcription to modern formats, indexing by graphical content, or even in forensics for writer identification.



(a) (b)

Figure 1.2: Examples of graphic documents: (a) Architectural drawing. (b) Old handwritten music score.

This thesis is focused on the identification of the writer in music scores, a particular scenario of graphic documents. Since music scores use a particular diagrammatic notation, let us describe the terminology and structure of a music score and the task of Optical Music Recognition.

Recognition of Music Scores

Music Scores are a particular kind of graphic document, which include text and graphic elements. The recognition of these documents has been a very active research topic field [BB92]. Optical Music Recognition (OMR) consists in the understanding of information from music scores (see an example in Fig.1.2(b)) and its conversion into a machine readable format.

Although OMR belongs to graphics recognition because it requires the understanding of two-dimensional relationships, OMR has many similarities with Optical Character Recognition (OCR), because whereas OCR recognizes characters in text, OMR recognizes musical symbols in scores.

The most common music symbols in a music score are notes, rests, accidentals and clefs (see Fig.1.3). Some terminology used in music notation is the following:

- Staff: Five equidistant, parallel, horizontal lines on which music symbols are written. They define the vertical coordinate system for pitches and provide horizontal direction for the temporal coordinate system.
- Attributive symbols at the beginning: Clef, Time and Key signature.
- Clef. A symbol usually placed at the left-hand end of a staff, indicating the pitch of the notes written on it.
- Bar lines: Vertical lines which separate every bar unit or measure.
- Notes. Notes are composed of head notes, beams, stems, flags and accidentals.

- Accidental. A sign indicating a momentary departure from the key signature by raising or lowering a note.
- Rest (pause). Interval of silence of specified duration.
- Slurs: Curves that join musical symbols.
- Dynamic and Tempo Markings indicate how loud/soft the music should be played, and the speed of the rhythm of a composition.
- Lyrics: The set of words that will sing the chorus or singers.

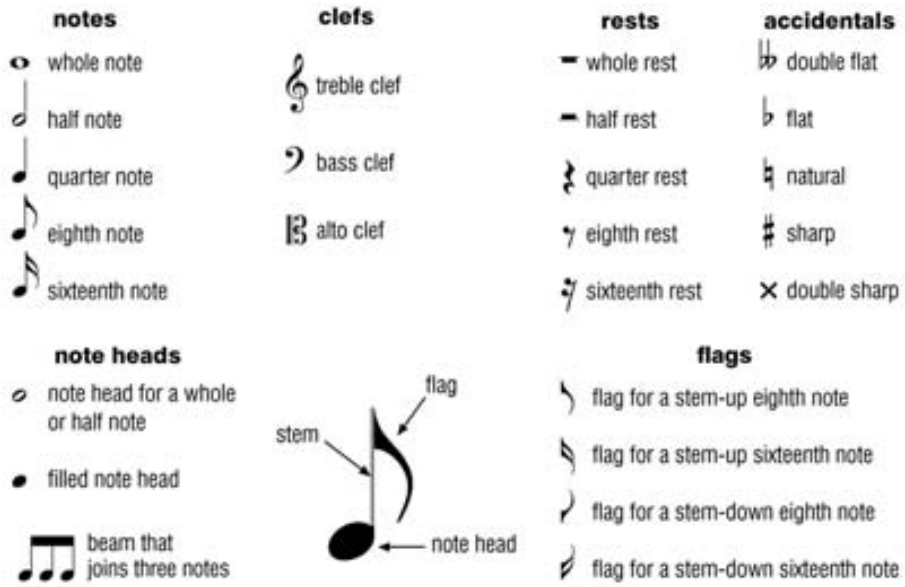
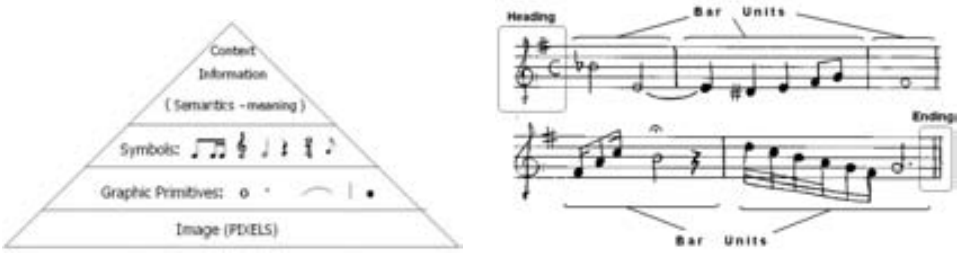


Figure 1.3: Common elements of Music Notation.

Similarly to OCR systems (which include the pixel, character, word and semantic level), the levels of the processed information of an OMR system are the image (pixels), graphical primitive, symbol and context information level (see Fig.1.4(a)). Context information helps to correct errors, and whereas dictionaries are commonly used in OCR, the formal music language theory is used in OMR.

For an OMR system, we can consider that a music score has three important elements: Heading, Bar units and Ending (see Fig.1.4(b)). Heading consists in the clef (alto, trebble or bass clef), the time signature (usually formed by two numbers that indicate the measure) and the key signature (flats, sharps or naturals, which indicate the tonality of the music score). Bar units are the containers of music symbols (the amount of music symbols depends on the time signature). Finally, the Ending is usually an ending measure bar and sometimes includes repeating marks.

After a brief overview of the basis of the music notation, let us discuss the main properties used for musicologists for discriminating the different handwriting styles.



(a) (b)

Figure 1.4: OMR: (a) Levels of a OMR system, (b) Structure of a music score.

1.2.3 Characterization of the Graphical Properties for Writer Identification in Music Scores

In order to find the discriminating properties between the different writers in music scores, the characteristic writing style in music notation must be deeply analysized. In the *eNoteHistory* writer identification project in music scores [Lut02],[GÖ3],[BIM04] the following set of characteristics were proposed: The distance between two staff lines, the position of the note stem relative to the note head, the length and inclination of note stems, and the shape of music symbols (such as clefs, flags and rests).

Similarly, scholars in musicology from the *Universitat Autònoma de Barcelona* mainly use the following aspects for writer identification: the shape of clefs, rests, time signatures, ending signatures and lyrics. Figure 1.5 shows some pieces of music scores (without staffs) extracted from different writers, in which the differences in the shape of clefs, notes and rests are very discriminant. The personal characteristics useful to discriminate each writer are described next.

1. Staff lines: In case the staff lines are written by hand, the following information can be extracted: The width of the staff lines, the distance between them and their straitness. See Fig.1.6 for an example of a staff drawn by hand. Unfortunately, an important amount of music scores do not have hand-drawn staffs, and no useful information can be extracted.

2. Music Notes: The shape of music notes can be very peculiar and can be used for characterizing the writer style. In this sense, the following properties are important. First, information about the shape (circular, eliptical...), size and location in the staff are useful. In case of non-filled headnotes, the shape of the loop can be analyzed, and also whether the loop is completely closed or not. In Fig.1.5 one can see that the headnotes of the writer Milans look like triangles, whereas Clausell tends to write circular headnotes. Concerning the beam, the lenght, slant and straitness of the beam can be used. Also, in case the beam and the headnote are not connected, the gap distance to its corresponding headnote is taken into account. Finally, the flag of eight and sixteenth notes are analyzed, extracting information about the flag orientation, shape, the distance between flag notes (in case of sixteenth notes), and also the

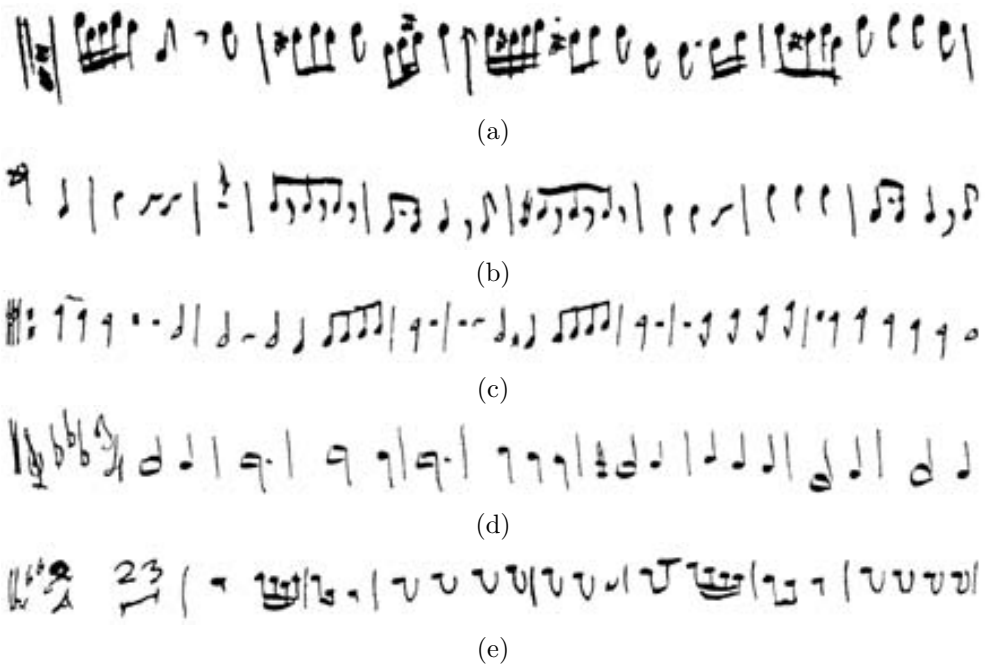


Figure 1.5: Pieces of music handwritings from different writers (composers): (a) Andreu, (b) Clausell, (c) Milans, (d) Aleix, (e) Sauri. One can easily notice the writing style differences in the shape of music notes (the half, quarter and eighth notes), flags, rests and clefs.

distance to the corresponding beam (in case they are not joined). See Fig.1.5(a),(c),(e) for noticing the differences in the curvature of the eighth flags.

3. Bar lines: The slant, length and straitness of the bar lines are also used, and also whether they cover the whole staff lines or not. In Fig.1.5(b),(c),(e) the differences in the slant of the bar lines are remarkable. It must be noticed that the differences between the bar lines of the different writers might not be enough important for ensuring a good discrimination power.

4. Rests: The shape of rests (pauses) is characteristic of the writer (see in Fig.1.5 the differences in the shape of the rests between the writer Clausell and Milans). In

Figure 1.6: Staff lines written by hand.

fact, when they appear in a music sheet, they are also very useful for determining the century of the composition.

5. Accidentals and Key Signature: The shape of accidentals (e.g. sharps, flats and naturals), the distance between their segments or their location related to the music note are useful (see in Fig.1.5(a),(b),(d) the differences in the shape of the sharps). The key signature is a group of accidentals that can appear at the beginning of a staff, and the distance between the accidentals can also be used for identifying the writer. Unfortunately, accidentals are not appearing in all the music sheets.

6. Clefs: Clefs are drawn usually at the beginning of each staff, indicating the pitch of the music notes. They are very useful for characterizing the writer style, because of the high variability of the music clefs. Notice the high variability of the same alto clef in Fig.1.5(a),(c),(e). For this reason, it can be seen as a signature of the writer, being very useful for identifying the writer of a music score.

7. Lyrics, Dynamics, Tempo Markings and Time Signature: Obviously, the handwritten text (lyrics) that appear in a music score characterizes the writer style, and lyrics can be treated as text, and consequently, used for writer identification. Dynamics and tempo markings (which indicate the speed and de strength of the interpretation) are letters or words (e.g. *f, p, mf, allegro, adagio*) that can also be treated as text identification. Also, the time signature (which indicates the measure) is composed of digits and letters, and can be also used for discriminating the writers.

8. Ending Signatures: In some music sheets, some writers used to draw a pothook at the end of the music score, as a personal signature (see Fig.1.7). The shape of these kind of strokes is very particular of a writer, and can distinguish among several writers, but unfortunately, these signatures are appearing in only a subset of music sheets, and could not be suitable for writer identification.

(a) (b) (c)

Figure 1.7: Ending signatures (in black) of three different writers.

In must be said that musicologists also perform a musicological analysis of the composition (melody, harmony, rhythm, etc.), because the music style is important to characterize a composer. In this field, some interesting research works have been done, such as the proposal of distinctive pattern features for music audio [CB08] and the music style identification approaches[CAVRPC$^+$03], [CP06]. Since the audio analysis is out the scope of this work, we will focus on the image analysis of the music score for writer identification.

1.3 Thesis Problem Statement

This thesis addresses the problem of writer identification in graphic documents, concretely, old handwritten music scores. Although writer identification in text documents has been subject of an intensive research, the identification of the author of a handwritten graphic document is still an open problem. Although some writer identification approaches used for logographic languages (such as the Chinese or Hebrew alphabet) could make use of graphic recognition methods, very few works are performed on graphic documents [BIM04].

Writer identification approaches for text can be classified in *text-dependent* and *text-independent*. In the first ones, a set of model word patterns is collected for each writer, and the identification is performed by a similarity function between such models and the unknown image. In the second ones, the meaning of the handwriting is unknown, increasing the difficulty but obtaining more general approaches. Concerning writer identification in graphic documents, text-dependent approaches can be renamed as *symbol-dependent* ones, because instead of recognizing text, they recognize hand-drawn symbols. This kind of symbol-dependent approaches make use of symbol recognition methods for recognizing the hand-drawn symbols that belong to the graphical language used in this particular kind of graphic document.

Hand-drawn symbol recognition is a particular case of handwriting recognition, which must deal with the variability among scripts and writer styles, or even between different time periods. For these reasons, commercial applications are usually constrained to controlled domains (such as bank checks or postal letters) that make use of contextual or grammatical models and dictionaries. The recognition of graphical symbols has two added difficulties regarding to handwritten text recognition. First, graphical symbols are bidimensional shapes appearing in bidimensional layouts, so, in addition to the distortions and deformations typically found in handwriting, the 1D models used for handwriting text recognition should cope with variations in sizes, rotation, and translation. Second, unlike text, graphical symbols can not easily benefit from the use of contextual and grammatical models (such as dictionaries used for text recognition).

Comparing to symbol recognition methods for printed documents, the difficulties of hand drawn symbol recognition methods increase. Firstly, because of the inherent distortions present in handwritten symbols (see Fig. 1.8), consisting mainly in inaccuracy in junctions, hooklets, circlets, elastic and anisotropic deformation in strokes, over-tracing, overlapping, gaps or missing parts. Secondly, the variability of symbol appearance is an important problem when the number of writers increases. In such cases the recognition approach must cope with the variability of symbol appearance because of the high differences in writer styles, with variations in sizes, shapes and pressure in strokes. See Fig. 1.9 for an example of the huge variability in music clefs' appearance. Secondly, the difficulty increases when the number of writers is unconstrained. In such cases the recognition approach must cope with the variability of symbol appearance because of the high differences in writer styles, with variations in sizes, shapes and pressure in strokes. See Fig. 1.9 for an example of the huge variability in music clefs' appearance (one can hardly believe that there are only three different classes).

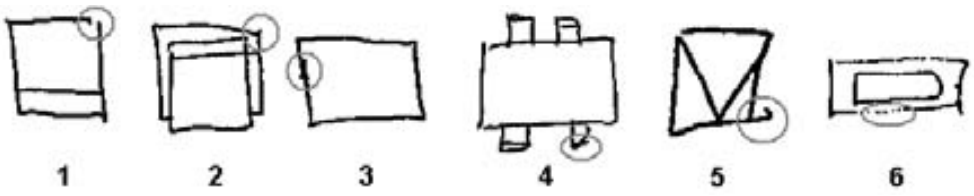
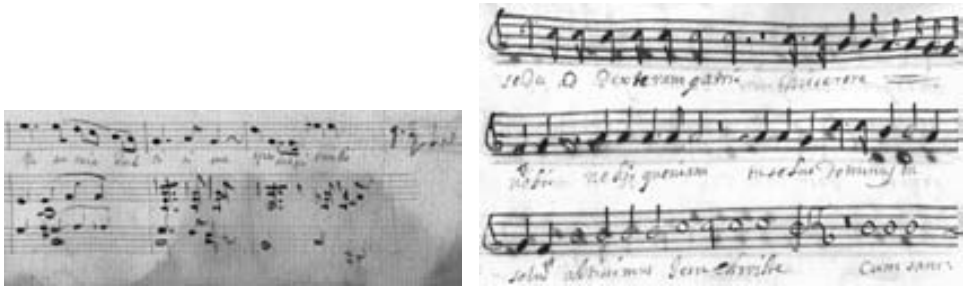


Figure 1.8: Distorted shapes: 1:Distortion on junctions. 2:Distortion on angles. 3:Overlapping. 4:Missing parts. 5:Distortion on junctions and angles. 6:Gaps

(a) (b) (c)

Figure 1.9: High variability of hand drawn musical clefs: (a)Treble, (b)Bass, (c)Alto

Concerning historical documents, we find some added difficulties: First, paper degradation (see Fig.1.10(a)) requires specialized image-cleaning and enhancement algorithms. Second, show-through and bleed-through problems can difficult the distinction between background and foreground (see Fig.1.10(b)). Third, in historical documents there is usually a lack of a standard notation, because notation differs from a century to another.



(a) (b)

Figure 1.10: Examples of paper degradation in old music scores: (a)Some sections of the staff lines are missing, (b)Some show-through problems and wholes provoked by ink.

Finally, as it has been discussed in Section 1.2, the access to historic music scores is only allowed to some expert musicologists. Since there is no public database of old handwritten music scores, the construction of a database of old handwritten music scores is required.

1.4 Objectives and Contributions

The main objective of the thesis is to perform **writer identification in graphic documents, concretely music scores**. Although some graphic documents contain text and graphics, the objective is to use only the graphic notation to identify the writer. Concretely, most music compositions in last centuries were sacred music, containing lyrics (text) for the chorus and the solists. In these scores, some writer identification methods for handwritten text documents could be applied for lyrics. However, the aim of this thesis is to use only music symbols to perform writer identification. Moreover, the methodology will also be useful for writer identification in those music scores that contain no text, such as music scores for instruments.

The research work above exposed can be divided in several goals and their corresponding contributions:

1. Proposal of a writer identification architecture for music scores.

   The first objective consists of a proposal of the architecture for writer identification in music scores. It must include the review of the state of the art methods for writer identification. It covers the study of text-dependent and text-independent methods for writer identification in text documents and also the study of the existing methods for graphic documents, including music scores.

   The main contribution of this thesis is the proposal of a writing identification architecture for old handwritten music scores. It is an hybrid architecture (see Fig.1.11), which combines three different writer identification approaches. The first one is a *symbol-dependent* approach, based on symbol recognition, which extracts features from music symbols. The second one is a *symbol-independent* approach which extracts features from music lines. The third one is another *symbol-independent* approach which extracts features from music texture images. The integration and ensemble of the different approaches is also performed. In the architecture proposed, the classification results obtained for each classifier are combined, so that the overall writer identification rate are increased.

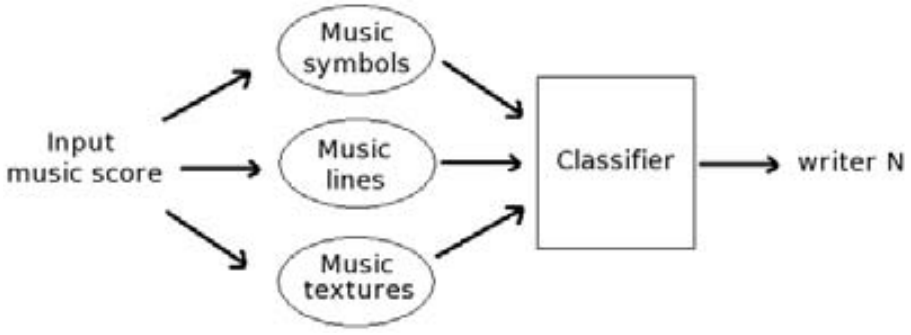


Figure 1.11: Writer identification architecture for music scores: features extracted from music symbols, music lines, and music texture images are combined for the final classification.

2. Study and proposal of symbol recognition methods for hand drawn symbols.

   A *symbol-dependent* writer identification approach requires the use of symbol recognition methods. A good symbol recognition method should make use of a symbol descriptor that guarantees intra-class compactness and inter-class separability. It should be tolerant to noise, degradation, occlusions, distortion and elastic deformations typically found in handwritten documents. After the study of the state of the art methods for symbol recognition, the following three symbol recognition methods for hand-drawn symbols have been proposed.

   - The first one is a Dynamic Time Warping-based symbol recognition method. It is a robust approach tolerant to writer style and hand drawn distortions. This method, which is invariant to scale and rotation, is based on the Dynamic Time Warping (DTW) algorithm. The symbols are described by vector sequences of specific features, and a variation of the DTW-distance is used for computing the matching distance.
   - The second method proposed defines the Blurred Shape Model (BSM) descriptor. This descriptor encodes the spatial probability of appearance of the shape pixels and their context information. As a result, a robust technique in front of noise and elastic deformations is obtained.
   - The third one is the Circular Blurred Shape Model (CBSM), which is an extension of the BSM to a circular grid. As a result, the descriptor can identify the rotation angles required to align two symbols, rotating the symbols in case it is necessary. Thus, the descriptor becomes rotation invariant.

3. Proposal of a writer identification approach based on symbol recognition.

   Similarly to the text-dependent approaches used for writer identification in text documents (which recognize several elements before the identification step), the goal is to propose a *symbol-dependent* method based on symbol recognition. It must include an study of the existing approaches in Optical Music Recognition (OMR), covering the staff detection, segmentation, and music symbol recognition. Since there are very few approaches in OMR that can cope with old handwritten music scores (even less approaches concerning old handwritten ones), a method for removing the staff, segmenting elements and extracting features from music symbols is required.

   The main contribution related to this objective consists in the proposal of the following methods for removing the staff lines and extracting specific features from music symbols:

   - Staff Removal. Before recognizing the music symbols for extracting features, an analysis of the music score is performed. As a contribution, a method for detecting and removing the staff lines in old handwritten music scores is proposed, coping with gaps, deformations due to paper degradation and the warping effect. The method consists in the use of horizontal projections, median filters and contour tracking.

- Detection of music clefs. Once the staff is removed, a method for segmenting and detecting clefs and music notes is proposed, which uses morphological operations and a combination of the BSM and DTW-based symbol recognition methods.
- Specific features for music symbols. Finally, a novel set of features is proposed for extracting information about the shape of the music clefs. For this purpose, the BSM descriptor is used for extracting information about the shape of the symbol.

4. Adaptation of writer identification methods for text to music scores.

   The study of the applicability and adaptation of writer identification methods for graphic documents is required. The adaptation of existing writer identification methods applied to text is a difficult task. Since graphical data is bidimensional, preprocessing steps are required to adapt this information before applying the existing writer identification techniques.

   The main contribution related to this goal consists in the adaptation of two writer identification methods to music scores. Two off-line text-independent approaches for performing writer identication in musical scores are adapted. Contrary to the symbol-dependent approach based on symbol recognition, the following two methods avoid the recognition of the elements in the score:

   - Music Lines. For each staff, a normalized music line is obtained. It consists in removing the staff lines and centering all the music symbols in a reference line. Afterwards, we extract 98 features based on basic measures, connected components, contours and fractal features. These features are extracted from the set of 100 features described by Hertel and Bunke in [HB03].
   - Music Textures. Image textures are generated from music symbols before applying textural features. Several methods for generating texture images from music symbols are proposed. Every approach uses a different spatial variation when combining the music symbols to generate the textures. Once the image textures are obtained, textural features such as Gabor filters and Grey-scale Co-ocurrence matrices [STB00], are computed.

5. Construction of the databases for the evaluation framework.

   Since there is no public database of old handwritten music scores available, the last goal is the construction of a framework for validating the proposed methodology for music scores.

   The main contribution related to this last objective is the construction of the following databases:

   - Old Music Scores. The first dataset consists in the digitization of old handwritten music scores (from the 17th to 19th centuries) from three different archives in Catalonia: the archive of the Seminar of Barcelona, Terrassa and Canet de Mar. It also includes the study of the convenient resolution and the required operations for allowing the image to be more

readable. In this sense, the processing of the image will cope with noise, degradation, transparencies and the warping effect. The database obtained can help not only in the preservation of these documents, but also in their diffusion and analysis, including binarization, optical music recognition, writer identification and style classification.

- Music Symbols. The second dataset consists of segmented music clefs and accidentals, which have been extracted from modern and old music scores. It has been constructed in order to validate the different hand-drawn symbol recognition approaches proposed in this thesis.

1.5 Thesis Outline

The structure of the dissertation is the following:

- The state of the art of writer identification, optical music recognition and symbol recognition methods is reviewed in Chapter 2. First, related work on writer identification methods in handwritten text documents is overviewed. Due to the application of writer identification methods to music scores, an overview of the related work in optical music recognition is also performed. Finally, symbol recognition methods for printed and hand-drawn symbols are reviewed.

- The **Dynamic Time Warping-based symbol recognition method** is defined in Chapter 3, which describes the symbols using vector sequences of specific features and computes the matching distance using a variation of the DTW algorithm.

- The **Blurred Shape Model** and the **Circular Blurred Shape Model** are defined in Chapter 4. These approaches encode the spatial probability of appearance of the shape pixels and their context information. The main difference between them is that the Circular Blurred Shape Model uses a correlogram structure.

- The three writer identification approaches for music scores are presented in the following Chapters.

  The first **writer identification approach based on symbol recognition methods** is proposed and developed in Chapter 5. It consists in detecting and segmenting the music clefs using a combination of the DTW and BSM symbol recognition approaches. Afterwards, the BSM features are extracted from every music clef.

  The **writer identification approach based on features extracted from music lines** is presented in Chapter 6. Firstly, the music sheet is preprocessed and normalized for obtaining a single binarized music line, without the staff lines. Afterwards, 98 features are extracted for every music line, including basic measures (such as slant and width of the writing), connected components, lower and upper contour of the line and fractal features.

The **writer identification approach based on features extracted from texture images** is proposed in Chapter 7. First, several approaches for generating texture images from music symbols are described. Every approach uses a different spatial variation when combining the music symbols to generate the textures. Afterwards, Gabor filters and Grey-scale Co-ocurrence matrices are computed to obtain the features.

- The ensemble architecture is presented in Chapter 8. First, the generic preprocessing, consisting in the binarization and staff removal of the music score is described. Afterwards, the classification and combination of the three writer identification approaches is fully described. Finally, the global results of the system are presented.

- Conclusions are presented in Chapter 9. Firstly, a summary of the main contributions is performed. Afterwards, we discuss about the proposed approaches and their corresponding experimental results. Finally, future work is exposed.

- Finally, the music databases created and the grammar for OMR are described in the Appendixes.

Chapter 2

State of the Art

In this chapter the main writer identification methods are overviewed. We also review the optical music recognition methods because we are applying writer identification to the framework of music scores. Finally, symbol recognition methods are overviewed, because they are required for recognizing the music symbols in the score.

This chapter covers the state of the art of writer identification, Optical Music Recognition (OMR) and symbol recognition. Section 1 presents work related to the identification of the writer in text documents, old documents and music scores. Next, Section 2 presents the state of the art of Optical Music Recognition approaches, describing the main techniques used for each stage of the recognition system (preprocessing, staff removal, symbol classification and validation) and also the main systems used for ancient and handwritten music scores. Finally, Section 3 addresses the work related to symbol recognition, which is divided in methods for printed symbols, hand-drawn symbols and symbols in real environments.

2.1 Writer Identification

Since this thesis focuses on writer identification, let us briefly introduce the handwriting recognition. Handwriting is a classical area of study, which covers the tasks of recognition, interpretation, identification and verification of documents. Handwriting recognition consists in transforming the image in its symbolic representation, whereas handwriting interpretation is focused on determining the meaning of the input (e.g. the address of a letter).

There are many works in the literature on the recognition of handwritten languages based on Roman alphabet (see surveys [PS00], [Bun03]). According to [Bun03], handwriting recognition techniques can be divided into the tasks of recognizing isolated characters (Intelligent Character Recognition - ICR), cursive words, and general text. The recognition of isolated characters, which is quite similar to symbol recognition, is a mature area of study [Liu07]. Efforts are nowadays focused in the recognition of words and full sentences. In word recognition there are three different approaches:

holistic, segmentation based and segmentation free. Holistic methods [MG99] do not require the segmentation of words, recognizing the whole word, but there must be few number of classes. Segmentation based methods [LG02] perform a segmentation of words into characters or graphemes. Segmentation free perform the segmentation and recognition at the same time, usually using Hidden Markov Models (HMM) [GB04]. Text recognition in a free context is still an open problem [BZB06]. Main existing segmentation based methods use trees for the segmentation of sentences into words, and then HMM for the recognition of these words [KFK02]. Segmentation free approaches usually use HMM, and in some cases, also grammars [BB08], [ZCB06].

Handwriting identification and verification are related. Whereas handwriting identification consists in determining the author of a piece of handwriting from a set of writers, handwriting verification consists in determining whether the handwriting is from a given author. In handwriting recognition and interpretation, the idea is to filter out the variations in handwriting style to determine the meaning, whereas in handwriting identification and verification, these variations in handwriting style are fundamental for the purpose.

In writer identification, the are two natural factors in conflict: individual characteristics (within-writer variability) and class characteristics (between-writers variation). The goal is to find optimal trade-off between intra-class compactness (minimizing individual characteristics) and inter-class separability (maximizing class characteristics). Finally, it must be said that in most of the cases, a writer identification system performs a search in a database with handwriting samples, returning a list of candidates for the handwriting query, and a human expert takes the final decision.

The main techniques used for writer identification and verification are briefly commented in next subsections. Afterwards, approaches for dealing with the identification of the writer in old documents and music scores will be described.

2.1.1 Writer Identification in Text Documents

Writer identification in handwritten text documents is an active area of study (see the surveys [PL89], [PS00], [Sch07b]). Traditionally, the off-line approaches for writer identification in text can be divided in text-dependent and text-independent, depending on whether the writer has to write a predefined text.

In text-dependent approaches [BYBKD07], the system compares the individual characters/words with the known trancription. Thus, the system requires a handwriting recognition step. In these approaches, the relation between writer recognition and identification is very close. The common elements used for extracting features for writer identification are: alignment (reference lines), angles, arrangement (margins, spacing), connecting strokes, curves, form (round, angular...), line quality (smooth, jerky), movement, pen lifts, pick-up strokes (leading ligatures), proportion, retrace, skill, slant, spacing, spelling, straight lines, terminal strokes... Notice that some of these elements (e.g. slant, baseline angle) are also used for handwriting recognition.

In text-independent approaches [BS07b], [HS08], the meaning of the text is unknown, avoiding the segmentation and recognition of words. Consequently the system will be faster and more robust, avoiding the dependence on a good recognizer. Some of the most common approaches are briefly commented next.

Writer identification by the analysis of words and numerals. Some works base the identification in the analysis of words or numerals (digits). Zois et al. [ZA00] process horizontal projection profiles on single words. In this approach, the projections are partitioned in segments, and features are computed. Leedman et al. [LC03] extract a set of eleven features from previously segmented digits, such as the height to width ratio, the number of end points and junctions, number of loops, slant, zero crossings, pixel densiy, center of gravity, etc. Afterwards, the classification is performed using the Hamming distance.

Writer identification by the analysis of text lines. Hertel and Bunke [HB03] propose a method for extracting a set of 100 features from text lines, including basic measurements (such as slant and width of the writing), connected components, enclosed regions, lower and upper contour of the line and fractal features. This work is an extension of the system proposed by Marti et al. [MMB01], in which a set of 12 features (basic measurements and fractals) is used. The inclusion of fractals as features has been inspired in the writer identification and authentication systems proposed in [BVSE97] and [SGV03]. In these approaches, fractals are used for distinguishing the legibility degree of the handwriting, and for characterizing the "shape" of the handwriting.

Schlapbach and Bunke propose two text-independent methods [Sch07a]. The first one is based on Hidden Markov Models (HMM) [SB07a], developing an individual recognizer for each writer. In their approach, each recognizer is an expert of the handwriting of one writer, because it has been trained with text lines from only one writer. First, the text is normalized, and the following set of features are extracted by a sliding window: the fraction of black pixels, the center of gravity, the second moment order, the position and orientation or the upper and lower pixels, the number of black/white transitions. The method has been tested on a database of 100 writers, reaching a identification rate of 97% (first ranked author - Top1), and 98% (the five first ranked authors - Top4). The second proposed approach is based on the use of Gaussian Mixture Models (GMM) [SB08]. The authors claim that GMM are less complex than HMM, avoiding the transcription of text lines (required in HMM during the training step) and the modeling of words or characters. GMM can be seen as a single-stage HMM, with one output distribution function. Thus, it only requires to train the parameters of this output function. Authors demonstrate that the GMM outperform HMM, reaching a identification rate of about 98% (Top1) and 99% (Top4).

Bulacu and Schomaker [SB04] propose the use of connected-component contours and edge-based directional probability functions, which is performed considering two edge fragments in the neighborhood of a pixel, and then computing the joint probability distribution of the orientations of the two fragments. In [BS07b] the authors add the allograph features to the above set of features, improving the identification rates. Their approach has been also successfully applied in arabic handwritings [BSB07]. The use of graphemes has been inspired in the work by Bensefia [BPH03], [BPH05], in which each handwriting is characterized by a set of invariant features. The authors claim that the probability distribution of grapheme usage is characteristic of each writer and can be computed using a common codebook of shapes (obtained by clustering result of segmentation).

There are also other approaches, such as the ones based in the directional element features for chinese writer identification [LWD06], the analysis of ink texture of pen [FBS02], the subdivision of the image in sub-images for searching repeated patterns [SV07], or the analysis of pixel grey levels for giving information about the speed and pressure of the writing [WSV03].

Writer Identification by the analysis of text as texture images. Some authors treat writer identification as a texture identification problem, demonstrating that textural features can be successfully used for writer and script identification [STB00], [HS08]. The idea is to generate a uniform texture image from text lines, and then, extracting textural features. The texture image generation method consists in performing a normalization of the text lines, in which orientation and spaces between words and margins have a predefined size, and completing text lines in case it is necessary. Then, random non-overlapping blocks are extracted from the image (see examples in Fig.2.1).

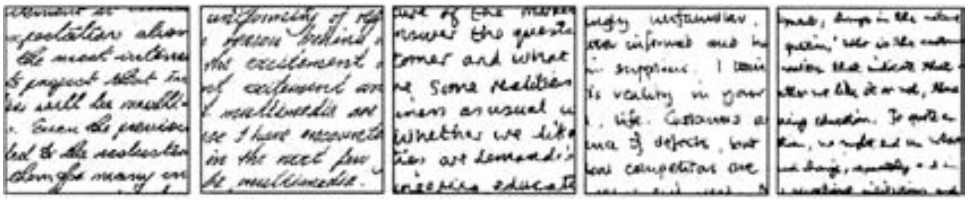


Figure 2.1: Examples of different handwritten images that can be perceptually classified as different textures (extracted from [STB00]).

Said et al. [STB00] use Gabor filters and Grey-Scale Co-ocurrence Matrices (GSCM) for extracting features from texture images. The authors have shown that these kind of features can be successfully applied not only to writer identification, but also to font identification in chinese and english documents [ZTW01]. They are inspired by the work by Tan and Peake [Tan96], [PT97] for script and language identification in printed documents, where seven different scripts (chinese, roman, greek, russian, korean, persian and malayalam) are identified using these features. In the experiments, they reach a identification rate of about 95%. In all these works, the Gabor features outperform the GSCM ones.

Other works have been proposed based on the wavelet transform to obtain features on the texture images, such as the following. He et al. [HYT+06] propose the use of a wavelet-based Generalized Gausian Density method. In [GBA07], Gazzah and Ben Amara propose the use of 2D Discrete Wavelet Transforms for writer identification in arabic handwritings. Hiremath and Shivashankar [HS08] use co-ocurrence histograms of wavelets for capturing information about the relationships between high and low frequency subbands. It is applied to script identification. In all these works, the authors compare their results with Gabor features, showing that wavelets obtain better identification rates.

Online writer identification Online writer identification is not as difficult as offline writer identification, because there is more information available, such as speed, order of strokes or pressure of the handwriting. Main approaches combine static information (offline) with dynamic features. Some works include the use of Gaussian Mixture Models [SB07b], the correlation between the length and direction of strokes [CF06], or the analysis of the velocity profile [CF07].

Writer Verification and Signature Verification

Whereas writer identification consists in identifying the authorship of a piece of handwriting, writer verification consists in determining whether two handwriting samples are written by the same person. The main difference is that in writer identification a list of matching writers is returned by the system, in writer verification, the output of the system is binary (accepted/rejected). Most of the approaches for writer identification can be easily used for writer verification, and some authors have also implemented a verification proposal from their identification ones [BPH05], [BS06], [SB08]. It must be said that Srihari et al. have developed some systems for writer identification [CS00b],[ZS03], they are mainly focused on writer verification systems, proposing the use of lexeme features [BSSS07], using dichotomy models [CS00a], statistical models [SBB+05], or even approaches applied to handwritten characters [SCAL02].

Signature verification is a special case of writer verification, whose main applications are the authentication of bank checks and biometric recognition (see the surveys [LP94], [PL89]). It has been a very active research field [PS00], appearing a lot of systems for both off-line [FAT05] and on-line data [CS07]. In a signature verification system, the scanned signature is compared with a few signature references provided by the user at the opening of the account. In these applications, the rejection of an authentic signature must be minimized, but even more important is the rejection of a forgery.

2.1.2 Writer Identification in Old Documents

There are some writer identification approaches applied to ancient handwritten text manuscripts, in order to perform historical analysis for classifying and identifying these documents. There are several techniques, including both text-dependent and text-independent approaches.

Text-independent approaches Eglin et al. [EBR07], [BEVA06] propose a text-indepentent and segmentation-free approach for writer identification in french manuscripts from the 18th century. First, they perform a binarization and noise reduction pre-process based on the Hermite tranform. Afterwards, they extract features only for the piece of handwritings with a minimum of five text lines and whose entropies (visual complexity/density of the image) are between certain thresholds, in order to perform a fair comparison between documents. A signature is computed for each image, which is expressed by a list of significant orientations (salient handwriting directions) and their corresponding Gabor densities. They obtain a function where the x-axis corresponds to the angular values and the y-axis corresponds to the Gabor

quantification. The comparison between signatures is performed using the Dynamic Time Warping algorithm.

Bulacu and Schomaker [BS07a] test their text-independent writer identification approach on medieval English documents (from 14th-16th centuries). Due to the complexity of the images (which also contain graphics), the authors manually select rectangular regions of homogeneous text. The method combines textural features (joint directional probability distributions), run-lengths information, and allographic features (grapheme-emission distributions). The authors are inspired by two previous works: the writer identification method proposed in [DS82] for ancient Hebrew writer identification, which uses run-length histograms; and the method proposed in [BPH03], which uses graphemes for writer identification in french documents of the 19th century. Bulacu and Schomaker use a feature fusion method, in which the distances are averaged and combined using the Hamming distance, improving the final classification rate. The method reaches a classification rate of 89% if the first ranked author (Top-1) is used, and 97% in case the first ten writers (Top-10) are considered.

Text-dependent approaches Yosef et al. [BYBKD07] describe a text-dependent approach for writer identification in Hebrew calligraphy documents from the 14th to 16th centuries. They also perform a sophisticated binarization technique for coping with degraded handwritings, which consists in a region growing scheme from the seed image of the characters. Afterwards, they detect and extract the three pre-specified letters using the morphological erosion operator. It must be said that in the hebrew alphabet, letters could be treated as symbols (see Fig.2.2). For all the specified letters found in the text, they compute a feature vector based on geometric parameters (such as the normalized central moment, aspect ratios, compactness, etc.). Finally, they use feature dimension reduction methods for increasing the final classification rate, reaching a 100% for 34 writers.

Figure 2.2: The three letters (Aleph, Lamed, Ain) used for writer identification in Hebrew documents (extracted from [BYBKD07]).

2.1.3 Writer Identification in Music Scores

The identification of the writer of music scores is still an open problem. To the best of our knowledge, only one project, the *eNoteHistory* [1] (*Scribe Identification in Handwritten Music Scores from the 18th Century*), has been performed about

[1] http://www.enotehistory.de

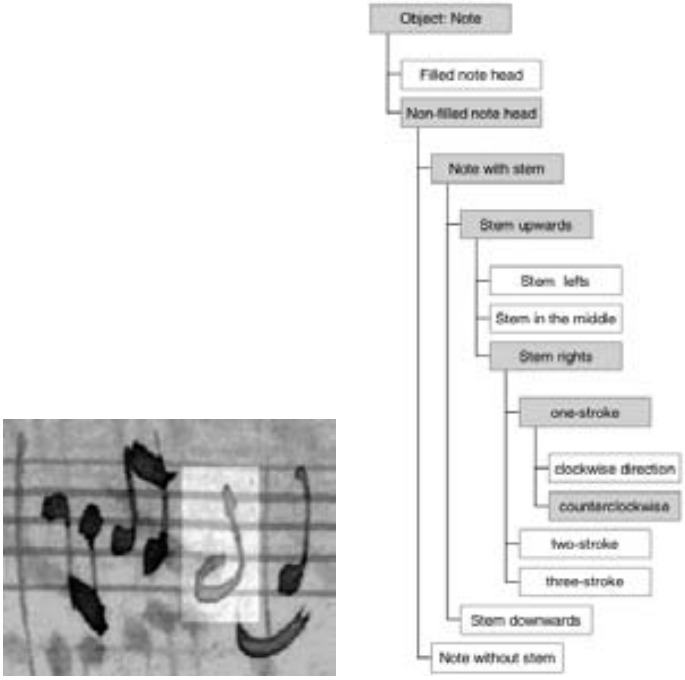
writer identification in music scores (see [Lut02], [GÖ3], [BIM04]). In this project, the researchers also claim that an important problem of the current registration of old historical music scores lays in the identification of the corresponding writer. The authors have developed a prototype that analyzes the music score and then extracts some features about structural information of the music symbols and notes. However no quantitative results have been published, and as far as we know, this work has not been continued.

Text-dependent approach The *eNoteHistory* is a text-dependent proposal for writer identification in musical scores which is based on the automated analysis of notation graphic features. The process of automated identification requires a definite level of handwriting content understanding. To extract a concrete feature of music objects means at first to recognize these music objects. Only after the recognition of separate note symbols from the whole note graphics, for instance, it is possible to describe them using characteristic feature sets.

Every note element (e.g. clefs, notes, rests...) will be represented by its tree structure, which shows an ideal appearance of the object (see an example of a tree structure for a half note in Fig.2.3). Each feature value node of the tree is represented by a textual description of a category, and in most of the cases, also by a pictogram. They use heuristically created distance matrices to compare two feature values, where each matrix includes a distance measure for each pair of possible values of a feature. The special stochastic rule-based method will be developed to handle an uncertain recognition results. The goal of the recognition problem is to relate the structures found in the image with the underlying object feature models. Once object features are given in the form of structural descriptions, the matching algorithm must solve the following three problems simultaneously: determine which image primitives belong to the same object feature, determine the identity of the structure and assign the correct object features to each image primitive.

The authors propose two different approaches to map documents in the feature space. The first one consists in a semi-authomatic procedure to allow musicologists to set the value of the features manually. The second one consists in performing an automatic approach for feature extraction. However, **the system deals only with the manual approach**, in which 150 feature sets are extracted. The classification is performed creating a cluster for each writer using the k-NN, and the Hamming distance is used for computing the distance between two feature sets. Results show a writer identification rate of 90% after optimization.

Concerning the automatic approach, only the staff removal and image analysis has been performed. The staffs are detected applying an edge detection (Sobel operator), horizontal projections and template matching with a template of five horizontal lines. Vertical line candidates are found using vertical projection, whereas note head candidates are found by a morphological closing and opening operator with a circle as structure element. Afterwards, they perform a template matching search for detecting the true note heads and stems (a stem always has an associated note head). The next two steps, namely object recognition and writer identification are still not implemented. For this reason, no results are shown in the papers.



(a) (b)

Figure 2.3: Example of the structural approach proposed in [Lut02]. (a) Half note, (b) Structural feature tree of the half note in (a).

2.1.4 Conclusions

In this section, we have presented the main writer identification approaches. Concerning the identification of text documents, there are two main groups, the text-dependent and the text-independent approaches. There are several text-dependent approaches that obtain good identication rates, however they involve the use of handwriting recognition methods which increases the difficulty. Contrary, text independent approaches are more flexible because they avoid the dependence of a good recognizer, keeping the performance in acceptable rate. In this subgroup of methods, there is an important amount of approaches which make use of different techniques, and none of them clearly outperforms the rest.

Concerning the identification of the writer in old music scores, we only find one relevant work in the literature, the *eNoteHistory* project [BIM04]. The published information about this project shows that the proposed approach is in a preliminar stage, and since there is no later publication, we may consider that the work has not been continued. Although no quantitative results have been published, in our opinion, the proposed approach shows promising discrimination power. Unfortunately, the proposed method requires an OMR for recognizing the most part of the music score in order to compute the features. Since an OMR for old music scores is an extremely difficult task (see Section 1.3), the proposed approach can not be developed so far.

2.2 Optical Music Recognition

Optical Music Recognition (OMR) is a classical area of interest of Document Image Analysis and Recognition (DIAR) that combines textual and graphical information. Since first works by Prerau and Pruslin in the 1960's, interest in OMR has grown in last decades, appearing several complete OMR systems and even an almost real-time keyboard-playing robot (the Wabot-2 [MSH89], see Fig.2.4). Some interesting surveys of classical OMR can be found in [BB92] and [Hom05], where several methods to segment and recognize symbols are reviewed. It must be said that most work through 1990 has focused on locating staffs and isolating and recognizing symbols. Nowadays, problems in OMR of printed scores include effective algorithms to interpret the resulting 2-D arrangement of symbols, and precise formalisms for representing the results of interpretation. Contrary to printed scores, few works have been done about the recognition of handwritten scores [Ng01], and ancient ones [CB92],[PVS03].

Figure 2.4: The WABOT-2 robot.

The input of an OMR system is usually a binarized image. Some advice about digitalization of music scores can be found in [RF03], in which the authors recommend scanning at a resolution of 600dpi, in at least 24-bit color, and storing images in PNG or JPEG file formats. Although most authors use an adaptive binarization technique, some authors [CB92] do not apply a binarization method because the scanner performs automatic thresholding to obtain a binary image. Others [Pre70] choose the threshold manually. In the vision system for the Wabot-2 [MSH89] the image is subdivided and each region is separately thresholded to allow for uneven illumination. The image is then rotated and normalized to compensate for distortions introduced in scanning. In [PBF07] some binarization techniques have been compared, including Otsu, Gatos, Niblack, Bernsen and Sauvola algorithms. Results show that depending on the degradation of the image, some techniques are better that others, but no one is the best in all kind of degraded images. Concerning noise reduction, in [CB92] a horizontal low-pass filter is used to remove short breaks in staff lines and symbols, whereas in [LC85] a three-by-three mask is used to eliminate isolated black pixels and to fill in isolated white pixels. Recent works ([Lut02]) use adaptive binarization techniques to binarize in a more robust way.

In this Section, a review of the research literature about OMR is presented. The recognition of a music score usually consists in the application of several functional stages. For this reason, we firstly review the main techniques used for each stage of the OMR system, consisting of the preprocessing and staff detection, symbol extraction and classification, and finally, validation. Secondly, techniques used for handwritten and ancient scores are reviewed.

2.2.1 Detection and Extraction of Staff Lines

Staff lines play a central role in music notation, because they define the vertical coordinate system for pitches, and provide a horizontal direction for the temporal coordinate system. The staff spacing gives a size normalization that is useful both for symbol recognition and interpretation: the size of musical symbols is linearly related to the staff space. Most OMR systems detect and remove the staff from the image in order to isolate musical symbols and facilitate the recognition process. Common staff removal methods take use of line tracking, runlenght analysis and vectorizations. Let us describe some of the most common techniques.

The approach proposed in [Pru66] eliminates all thin horizontal and vertical lines, including many bare staff-line sections and stems. This results in an image of isolated symbols, such as note heads and beams, which are then recognized using contour tracking methods. This preprocessing step erases or distorts most music symbols.

Prerau [Pre70] divides the process in fragmentation and assemblage. In the fragmentation step, the system scans along the top and bottom edges of staff lines to identify parts of symbols lying between, above and below the staff lines (a new symbol fragment is begun whenever a significant change in slope is encountered). In the assemblage step, these symbol fragments are connected if the two symbol fragments (separated by a staff line) have horizontal overlap. One disadvantage with this technique is that symbols which merge with staff lines do not always have horizontal overlap, so with this method, would keep disconnected when it should be connected.

Mahoney (see [Mah82]) uses a strategy similar to the symbol identification method: staff-line candidates are constructed (including all thin horizontal lines in the image) and the staff line descriptor (specifying allowable thicknesses, lengths and gap-lengths) is used to classify staff lines. The method removes only those parts of the line that do not overlap other symbols. Good extraction of staff lines is achieved, although more work is needed for dealing with line-region overlap.

Carter and Bacon (see [CB92]) propose a system for segmentation that uses processing based on a Line Adjacency Graph (LAG). Because the detection of places where a thin portion of a symbol tangentially intersects a staff line is difficult, mostly methods create gaps in symbols. Carter proposes a LAG-based analysis that successfully identifies such tangential intersections of symbols with staff lines. In addition, the system locates staff lines despite the image rotation of up to 10 degrees, copes with slight bowing of staff lines and with local variations in staff-line thickness. This method also uses horizontal projections to first locate staff lines.

In [KI91] the detection and extraction of staff lines is performed using histograms, run-lengths and projections. After determining the spacing of staff lines and their location (using run-lengths and histograms), the staff is analyzed (tracking from the

left), eliminating short horizontal runs whose width is under a certain threshold.

In [LC85] and [Dan98], projection methods are used to recognize staff lines, which are found in a Y projection. A defined threshold is used to select projections strong enough to be candidate staff lines. These candidates are searched to find groups of five equally-spaced lines. A score after staff removing is shown in Fig. 2.5.

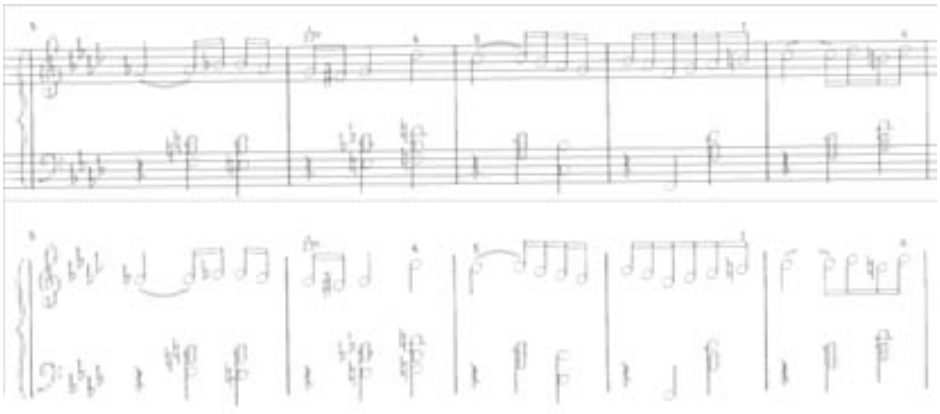


Figure 2.5: Staff removal image extracted from [LC85].

In [CBM88] the staff lines are located by looking for long horizontal runs of black pixels. Then the neighborhood of each staff-line pixel is examined to determine whether a music symbol intersects the staff line at this point. Staves are located by examination of a single column of pixels near the left end of the system. Large blank sections indicate gaps between staff lines, and are used to divide the image into individual staffs. Complete staff separation is no always achievable, because parts of symbols belonging to the staff above or the staff below may be included.

In [RCF+93] the method proposed consists in a vertical projection, projection filtering, local minima regions finding (those regions correspond probably to the regions where there are only staff lines), horizontal projection of each local minima regions (to ensure that that those peaks are certainly lines), and linking different peaks between them in order to built up the every staff. This technique is very robust because even if these lines are bowed, skewed of fragmented they are always found. For deleting staff lines, the thickness of each line is first estimated in according to the width values of the lines peaks. Then if width is smaller than a threshold (proportional to the estimated thickness) the line points at this place are erased. The problem of split symbols will be carried over in recognition stage.

Leplumey et al. [LCL93] present a method based on a prediction-and-check technique to extract staffs, even detecting lines with some curvature, discontinuities and inclination. After determining thickness of staff lines and interlines using histograms and run lengths, some hypotheses on the presence of lines is done grouping compatible observations into lines. Afterwards, an interpretation graph is used for joining segments to obtain staff lines. This method process allows little discontinuities thanks to the use of a local predicting function of the staff inclination.

The method proposed by Fujinaga in [Fuj04] detects staffs by horizontal projections and deskews each staff in the image. Afterwards, black vertical runs larger than a threshold are removed, and considers all remaining connected components with a considering width.

In [RT88] an OMR system for handwritten scores is described. In such scores, only musical symbols are drawn by hand, because staff lines are printed. The segmentation stage detects staff lines using measures of line angle and thickness. A window is passed over the image to compute a line-angle for every black pixel. The line angle is measured from the center of the window to the furthest black pixel in that window; this furthest black pixel is chosen so that the path from it to the center does not cross any white pixels. To detect staff lines, a large window radius is used. This causes covered staff-line sections to be labelled with a horizontal line-angle despite the interference of the superimposed musical symbols. Once a line angle has been determined, a line-thickness can be measured. These two measurements, combined with adjacency information are used to identify horizontal lines. The OMR system for handwritten recognition exposed in [RT88] shows acceptable performance results. Due to the low resolution of digitized images, it is difficult to estimate how this method would compare to others when applied to higher-resolution input.

An interesting comparative study of different staff removal algorithms can be found in [DDPF08]. The authors also proposed a new method consisting in the use of the skeleton of the image and obtaining staff segment candidates. A detection of false positives is also performed in order to improve the results. The authors have created distorted images from ground-truth data to compare their approach with existing staff removal methods. The authors conclude that there is not an algorithm that performs best in all kind of deformations. In fact, the proposed approach is quite robust in most deformations, but it is sensible to deformations that emulate historic prints.

2.2.2 Symbol Extraction and Classification

First of all, it must be said that some systems do not remove staff lines. The method proposed in [Pug06] uses HMM without any segmentation, and will be described in Section . The Wabot-2 robot [MSH89] performs a template matching without removing staff lines: staff lines are detected and used to normalize the image, to determine the score geometry, and also to restrict the search area for music symbols (then, the recognition of musical symbols must learn symbols which include segments of staffs). Staff lines are detected in hardware by a horizontal line filter, tolerating some skew. Where five equally-space lines are found, a staff is deemed to exist. Normalization parameters include staff location, staff inclination, area covered by staff and note-head size. Afterwards, the image of each staff is normalized according to these parameters.

For the classification of music symbols, different techniques have been proposed. Some of them are described next.

Pruslin [Pru66] uses contour tracking to describe connected binary image regions which remain after deleting horizontal and vertical lines. Classification depends both on stroke properties as well as on inter-stroke measurements (a method for template matching using contour strokes is developed).

In [Pre70], relative symbol size is used for an initial classification. The bounding-box dimensions are expressed in staff-space units. Dimensions of the bounding box are used to look up a list of possible matches (there is a pre-computed table containing the standard areas of each symbol in a height/width space). Typically there are three to five possible matches for each symbol, so heuristic tests are used to distinguish symbols that overlap in the height/width space, taking advantage of the syntax, redundancy, position and feature properties of each symbol type. Notice that, this classification is dependant of the publisher, and will not work in handwritten scores.

In [Mah82] pattern primitives (such as note heads, stems beams and flags) are combined to form music symbols (e.g. notes, chords and beamed note sequences). It does not use context information for the recognition of primitives, but it is used to infer musical symbols from the relationships between the various kinds of primitives. After extracting line primitives, dot primitives are processed and removed. All measures of distance are normalized on staff-line and staff-space thickness. Sample line parameters are principal direction, angle, thickness, length and maximum permitted gap. Sample region parameters are mass, width, height and inclination angle. This process is initially used in an interactive mode to add or modify object descriptions.

In the Wabot-2 robot [MSH89], musical symbols are recognized according to a two-level hierarchy: the upper level (in which the recognition of staff lines, note heads and bar lines is done) is implemented in hardware and the lower level in software. The search is performed using hardware-implemented template-matching.

Lee and Choi ([LC85]) use projection methods to recognize staff lines, bar lines, notes and rests. Once an image containing only a staff nucleus is obtained, an X and Y projections are used to find bar lines. Notes are recognized using X and Y projections from a small window around the symbol. Characteristic points in the projections are used for classification; a comparison is made with stored projections for known symbols. The main disadvantage of this method is that it is rotation-sensitive.

In [CBM88] an initial classification is obtained from the symbol height and width (as in [Pre70]), and then pixels in few particular rows and columns of the symbol-image are examined (because complete template matching is too computationally expensive). Some preliminary work on chord recognition is also present. An important problem is the noise-sensitivity of the method.

In [MN95] the extraction of heads and stems in printed piano scores is performed using a neural network: After extracting all regions candidates of stems or heads, a three-layer neural network is used to identify heads; the weights for the network are learned by the back propagation method. In the learning, the network learns the spatial constraints between heads and surroundings rather that the shapes of heads. Afterwards, this networks are used to identify a number of test head candidates. Finally, the stem candidates touching the detected heads are extracted as true stems.

In [MRHS93] the recognition system for printed scores is composed of two modules: the low-level vision module uses morphological algorithms for symbols detection; the high-level module context information to validate the results. Because morphological operations can be efficiently implemented in machine vision systems, the recognition task can be performed in near real-time.

In [RT88] knowledge about music notation is represented in a rule-based system, which is applied starting with the earliest steps of symbol segmentation and recogni-

tion. Images are digitalized in low resolution, so it is not clear the effectiveness of the system. The primitives recognized are: circular blobs, circles, horizontal lines, non-horizontal line segments and arcs (clefs are not recognized). Primitive identification is coded as several steps, using context information in the last step. Note-head detection is extremely difficult in these handwritten images, and a general-purpose blob detector does no work. Thus, note heads are searched for in constrained locations: first, verticals lines are located, then a thickness measure is used to test for wide spots at the ends of each potential stem; if there is a wide spot (whose circularity is under a certain threshold), it is accepted as a note head.

In [BC97] and [CB92] the recognition system is in an early stage. Objects are classified according to the bounding-box size, and according to the number and organization of their constituent sections (see an example in Fig. 2.6).

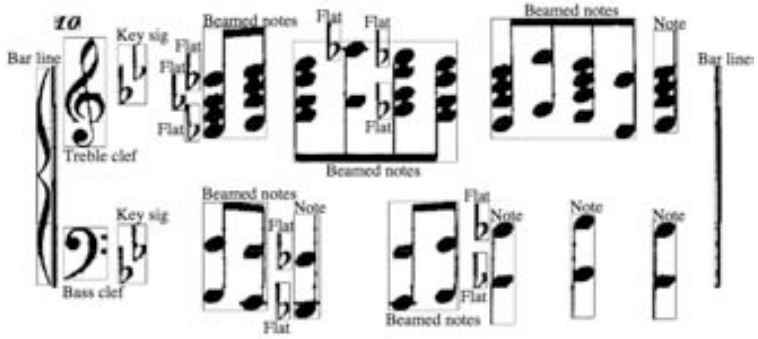


Figure 2.6: Classification of musical symbols performed in [BC97].

In [SD99], a probabilistic framework for recognition of printed scores is presented. The system uses an explicit descriptive model of the document class to find the most likely interpretation of a scanned document image, carrying out all stages of the analysis with a single inference engine (which allows for an end-to-end propagation of the uncertainty). The global modeling structure is similar to a stochastic attribute grammar, and local parameters are estimated using Hidden Markov Models (HMM). The HMM parameters are estimated from a training set using the segmental k-means algorithm. It also uses The Hough Transform and a low-band filter to locate lines and note heads of note groups.

In [RCF+93] symbols are isolated by using region growing method and thinning. After the polygonalization of the object, a parameter defined as a minimum distance to the contour is attributed to each segment, so spurious segments can be eliminated and some configuration segments are transformed. Once the skeleton structure is simplified, the attributed graph is constructed: the graph nodes correspond to the segments and the graph arcs to the links between segments. After constructing the graph, symbols are classified in symbols including notes with black heads (there is at least one segment having a distance to the contour exceeding a certain threshold) and the others. Half notes are detected if there is a stem with a little loop in its extremes.

In [CR95], a grammar is formalized to work in the image level to produce an accurate segmentation and thus accurate recognition. Whereas most grammars are

usually used at a high level to validate the structured document, the system proposed uses context information (syntax) to control the entire recognition process.

In [KI91] a sophisticated symbol recognition stage with a top-down architecture is performed. It uses a collection of processing modules which represents information about the current bar of music at five levels of abstraction: pixel, primitives, music symbols, meaning (pitch and duration of notes) and context information (interpretations). The four processing modules (primitive extraction, symbols synthesis, symbol recognition and semantic analysis) are made up of one or more recognition and verification units. The primitive extraction module contains units for recognizing stems, beams and note heads. Hypothesized primitives are removed from the pixel image. Unacceptable hypotheses are rejected at higher layers, are sent back to lower layers for further processing. Symbol recognition proceeds one measure at a time, and consists on pattern processing and semantic analysis (using context information), required for solving ambiguities of complex notations.

In [RB05] a system based on a fuzzy modeling of symbol classes and music writing rules is proposed. First, the individual analysis process (based on pattern matching) performs the segmentation of the objects and the correlation with symbol models stored in a reference base. Then, the fuzzy modeling part provides for each classification hypothesis a possibility degree of membership to the class. It also introduces a fuzzy representation of the common music writing rules by expressing graphical and syntactic compatibility degrees between the symbols. The fuzzy modeling of symbol classes allows to deal with imprecision and variations of symbol shapes.

Homenda and Luckncr [HL06] present a system for recognizing five different classes of music symbols. They compare methods based on centroids, Zernike moments and decision trees with split decision. They propose decision trees based on the linear combination of 278 basic features (e.g. histograms, density, symbol direction). They use Principal Component Analysis for improving the final recognition rate.

In [TSM06] a symbol recognition method for printed piano music scores with touching symbols is presented. The symbol candidates are detected by template matching, and from these candidates, correct symbols are selected by considering their relative positions and mutual connections. Touching primitives are detected using coherence check.

2.2.3 Validation

Rules on music notation makes the recognition task easier, because the information of two-dimensional relationships between musical symbols can be captured in a syntactic description of music. For that reason, most authors define grammars describing the organization of music notation in terms of music symbols. Some authors [AA82], [Pre70] use two different grammar levels: lower level grammars for music symbols (with terminal geometric figures such as dots, circles and lines and adjacency operatiors such as above, below, right of...) and high level grammars for music sentences (larger units with measures containing music symbols).

In [MSH89] the robot uses a musical grammar to correct errors such as missing beats or contradictory repeat signs. Examples of constraints applied to three-part organ music are: a fat double bar appears only at the end of each part, a treble or

bass clef always appear right at the start of each staff, the number of beats in each measure should match the time signature, etc.

In [MRHS93] a high-level reasoning module is developed. The system utilizes prior knowledge of music notation to reason about spatial positions and spatial sequences of recognized symbols. This module is composed of a connected components analysis and a reasoning module (that verifies if every musical symbol accomplishes its own constraints). The high-level module also employs verification procedures to check the veracity of the output of the morphological symbol recognizer.

In [FB93] a graph grammar for recognizing musical notation is presented, where the input graph to the grammar is constructed as a set of isolated attributed nodes representing the musical symbols. The grammar itself forms the edges representing the significant associations between the primitives that are necessary in determining the meaning. Although the proposed approach relies on the ability to control the order of application of the productions, there may be some portions of the grammar in which the order need not be specified, so, potential parallelism in the grammar is also made explicit.

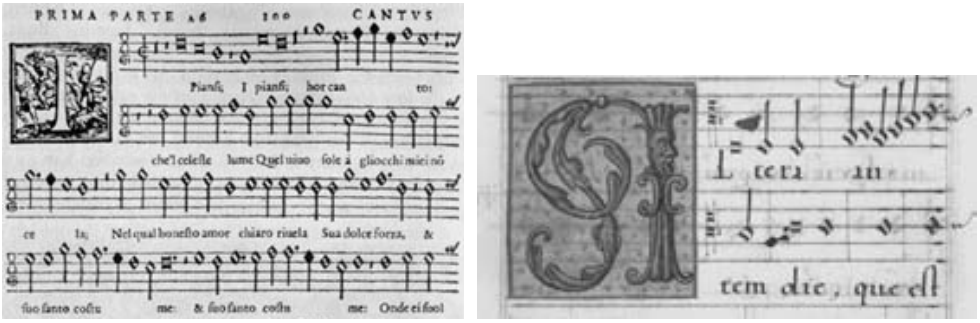
2.2.4 OMR for Ancient Music Scores

Bainbridge and Carter's system In [CB92], [BC97] and [BB96] a system based on a Line Adjacency Graph (LAG) is exposed. This system is also used to recognize ancient scores [Car95], consisting in scores of madrigals (see Fig. 2.7(a)) notated in *White Mensual Notation*. Symbols are correctly segmented and an early classification stage has been implemented.

This system successfully identifies tangential intersections of symbols with staff lines, locates staff lines despite the image rotation of up to 10 degrees, copes with slight bowing of staff lines and with local variations in staff-line thickness. Region information, derived from the LAG, is used to determine whether a symbol has merged with a staff line. The LAG is formed directly from a vertical run-length encoding of a binary image. A transformed LAG is formed by linking together neighboring segments to form sections. Junctions occur when a segment in one column overlaps several segments in an adjacent column; sections are terminated at these junctions. In the transformed LAG, each section is represented by a node in a graph, and junctions are represented by edges in the graph. The nodes in the transformed LAG should correspond to structural components of musical symbols. Then, the transformed LAG is searched for potential staff-line sections (filaments): sections that satisfy criteria related to aspect ratio, connectedness and curvature. Collinear filaments are concatenated together into filament strings. After staff lines are identified, the transformed LAG is restructured: further merging of non-staff sections takes place, now that junctions with staff staff-line sections have been specially marked. At this point, musical symbols are effectively isolated from the staff lines. Connected non-staff-line sections are combined to form objects, which correspond to music symbols or to connected components of music symbols.

Concerning the classification stage, the system performs a description of objects which correspond to music symbols or connected components of music symbols. These segmentation results are interpreted by a recognition system, where the objects re-

sulting from the segmentation are classified according to bounding-box size, and the number and organization of their constituent sections. The author comments that if there are overlapping or superimposed symbols another algorithm will be required.



(a) (b)

Figure 2.7: Examples of ancient musical scores, extracted from [Car95], [PVS03]

ROMA: Ancient Music Optical Recognition In [PVS03], a OMR method to recognize ancient musical scores (see Fig. 2.7(b)) is described. This system copes with specific notation of ancient documents, and is developed under the Portuguese project ROMA (Ancient Music Optical Recognition).

After the preprocessing stage, the segmentation module divides the music sheet in staff lines, bars and musical symbols. The staff lines are identified using horizontal projections and small rotations of the image. Then, segments of line whose thickness is not bigger than a certain threshold are removed. Bar lines are located using vertical projections, and objects are segmented using morphological operations and connectivity analysis.

The recognition process is based on a graph structure of classifiers, divided into two steps: feature extraction and classification. The method includes the construction of a class hierarchy associated with recognizers that distinguish between clusters of classes based on selected object features. Then, a method for the search of optimal graph hierarchy (manual and automated) and for the classification algorithms themselves is proposed. Finally, the reconstruction stage is needed to relate the recognized symbols with each other and with its staff lines and bars, creating the final description of the music. The system proposed obtains high performance results (97% of accuracy).

Pugin et al. method for OMR in old scores An approach for OMR in printed scores from the 16th-17th is presented in [Pug06]. The system consists in a segmentation-free approach based on Hidden Markov Models (HMM). They not remove the staff lines, and they do not perform any segmentation neither. The goal is to avoid segmentation problems and irregularities. The modeling of symbols on the staff is based on low-level simple features, which include the staff lines (see Fig.2.8). For feature extraction, they use a sliding window as in speech recognition, extracting the following 6 features for each window: the number of connected black zones, the

gravity centers, the area of the largest black element and the smallest white element, and the total area of the black elements in the window. Concerning the HMM, the number of states used matches as closely as possible the width in pixels of the symbol. The training is performed with the embedded version of he Baum-Welch algorithm. For every training iteration, each staff is used once to adapt the models corresponding to the symbols of which the staff is made. The author shows that with the use of these features and HMM, good recognition rates are obtained.

Figure 2.8: Examples of symbols with staff lines, extracted from [Pug06]

The authors also compare two OMR approaches (Gamut and Aruspix) applied to ancient scores in [PHBF08]. The authors claim that although Aruspix HMM models outperform the Gamut kNN classifiers, experiments show that paper degradation affect to the performance of both systems. The authors also perform an evaluation of binarization techniques for OMR in [PBF07], which has been already commented in the subsection of Binarization and Noise Reduction.

2.2.5 OMR for Handwritten Modern Music Scores

Kia Ng exposes in [Ng01] a prototype for printed scores, followed by a prototype for handwritten ones, discussing the limitations of the first one for handwritten scores processing. In the printed one, after binarizing and correcting the skew of the image, staff location is obtained using horizontal projections, and a line tracing algorithm with a local vertical projection window. Afterwards, a sub-segmentation algorithm disassemble musical symbols into graphical primitives, and the classification stage begins: first, isolated primitive musical symbols, clef and time signature are recognized. The recognition of other primitives is performed by interplay between the classification and the sub-segmentation modules (symbols not recognized are subdivided depending on its orientation). The classification module uses the aspect ratio of a bounding box, using a k-NN classifier. The prototype for printed scores recognizes 12 different sub-symbols with a 95% reliability.

Concerning the prototype for handwritten scores, the skeleton of the binary image is obtained in order to transform musical symbols into a set of interconnected curved lines. Then, junction and termination points are extracted from the skeleton representations. In the staff detection phase, all horizontal line segments are parsed to determine if they belong to part of a staff line using basic notational syntax and an estimated staff line height.

An additional process using a combination of edges, curvature and variations in relative thickness and stroke direction is used to perform further sub-segmentation and segregate the writings into lower-level graphical primitives (lines, curves and ellipses). Afterwards, primitives are classified using a KNN classifier. Each terminal point is parsed to search for any other nearby terminal points which are collinear with the current segment or following a polynomial extrapolation from the terminal points of

the current segment. The author comments that a tracing routine using a database of isolated handwritten musical symbols would improve the classification stage.

After the classification phase, these sub-segmented primitives are regrouped (applying basic syntactic rules) to form musical symbols. Contextual ambiguities are resolved using relative positions of primitives in the staff, and between primitives. The reconstruction module offers an intermediate stage where extensive heuristic, musical syntax and conventions could be introduced to enhance or confirm the primitive recognition and re-groupings. Unfortunately, no recognition rates are shown in the recognition of handwritten scores.

Online OMR Finally, an online symbol recognition system [MM07] must be also commented. The authors propose two kind of features for recognizing strokes: time-series data and hand-drawn image features. Then, features are combined to identify the music symbol. An eight-direction Freemain Chain Code is used to represent the time-series data of the stroke, and for matching the codes, string edit distance based on Dynamic Programming is used. For the computation of the image features, the image of the stroke is divided into $8x8$ regions, and the directional feature of each region is calculated. Then, a Support Vector Machine is used for the classification. Results of both classifiers are also combined using a Support Vector Machine. Afterwards, the combination of specific strokes for each music symbol is consulted in a pre-defined table. To allow a stroke distortion, some music symbols have several possibility combinations of strokes.

2.2.6 Conclusions

In this section, main OMR systems have been described. Table 2.1 shows main techniques used in staff detection and removal, whereas table 2.2 shows the main techniques used in classification of musical symbols. The validation phase is performed basically using rule-based reasoning, i.e. syntactical approaches that model the valid scores by a grammatical formalism. Finally, results of main systems exposed should be commented.

The system defined in [Pru66] only recognizes quarter notes, beamed note groups and chords, whereas Prerau's approach [Pre70] recognizes a more complete set of symbols (clefs, accidentals, half quarter and eight notes) with good recognition rates. Andronico [AA82] describes a system that recognizes clefs, key signatures, notes, rests and accidentals in simple monophonic music. In [Mah82] a system with human interaction recognizes simple polyphonic music, whereas Clarke's system [CBM88] recognizes single line melodies with a 90% accuracy.

The Wabot-2 can perform fast, accurate recognition of simple three-part organ scores, recognizing notes, clefs, accidentals, time signatures, bar lines, beams, rests, staccato and marcato marks, but it does not recognize words, slurs, ties, expression marks, ornaments and tempo indications.

In [LC85] the system exposed recognizes staff lines, bar lines, notes, chords and rests. Carter [CB92] has developed a system that segments under difficult imaging conditions, without an excess of ad hoc rules. It recognizes solo instrument parts, solo instrument with piano accompaniment and orchestra score with good tolerance

Staff Detection: Author	Techniques
Prerau	Contour Tracking
Mahoney	Construction of candidates, Staff line descriptors
Carter and Bainbridge	Projections, LAG
Kato, Lee, Dan, Clarke, Randriamahefa	Histograms, Runlengths, Projections
Leplumey	Runlengths, Reconstruction using a Graph
Roach	Slide-window, Orientation of line segments
Dalitz	Skeleton, Reconstruction using orientations of segments

Table 2.1: Main Techniques used in Staff Detection

Classification: Author	Techniques
Prerau	Bounding-box, Matching
Mahoney	Features of primitives, Descriptors
Carter and Bainbridge	Bounding-box, LAG
Kato	Pattern processing, syntax analysis
Lee	Projections
Clarke	Bounding-box, pixel analysis
Vuilleumier	Hidden Markov Models
Randriamahefa	Polygonalization, Attributed Graph
Ng	Skeletons and Mathematical Morphology Operators
Couasnon	Grammars
Toyama	Template Matching
Homenda	Decision trees

Table 2.2: Main Techniques used in Classification of musical symbols

to noise, limited rotation, broken print and distortion. The system exposed in [KI91] handles complex music notation (including two voices per staff with chords and shared note heads, slurs and pedal markings) with high performance rates. The OMR system for handwritten recognition exposed in [RT88] shows acceptable performance results. Due to the low resolution of digitized images, it is difficult to estimate how this method would compare to others when applied to higher-resolution input.

The grammar formalized by Couasnon in [CR95] can recognize notes, rests, chords, accents, clefs, key and time signature, phrasing slurs, dynamic markings. Abbreviations, ornaments and lyrics are not included. The classification system for the symbols and the segmentation and merging of connected components is under development, and no performance results are shown. Concerning the system proposed in [PVS03] for old scores, obtains high performance results (ancient scores are recognized with a 97% of accuracy). The output of the system is a normalized music sheet with the original staff printed using straight staff lines and normalized symbols.

The prototype for printed scores described in [Ng01] recognizes 12 different subsymbols with a 95% reliability. The output is expMIDI, which is compatible with the standard MIDI file format, and capable of storing expressive symbols such as accents and phrase markings. No recognition rates are shown in the recognition of handwritten scores. The online music recognition system proposed in [MM07] reaches a recognition rate of 98%.

As a summary, we can note that the recognition of printed music scores is a mature are of study (in which several approaches obtain very high recognition rates). Concerning the recognition of old printed ones, there are some existing works obtaining good results, although further work should be performed. Contrary, very few works have been done about the recognition of handwritten ones, being still an open problem.

2.3 Symbol Recognition

Symbol recognition is one of the main topics of Graphic Recognition, which has been an intensive research work in the last decades [LVSM02]. Symbols are synthetic visual entities made by humans to be read by humans. They are a good way to express ideas, allowing users to describe complex models with compact diagrammatic notations. The alphabet of symbols that belong to these diagrammatic notations is identified and interpreted in the context of a domain-dependent graphic notation.

Symbol recognition can be applied both to documents and real images (see examples in Fig. 2.9). Among the typical applications of document analysis, we can find the following: the analysis and recognition of logical circuit diagrams, engineering drawings, maps, architectural drawings, musical scores, or even logo recognition. On the other hand, detecting symbols from real images involves a large number of applications: recognizing logos with PDA cameras and smartphones, searching by symbolic queries anywhere, image retrieval, driving assistance (traffic signs) and blind person aid systems.

Figure 2.9: Examples of symbols that can appear in documents and real images.

The most typical visual cues for recognizing symbols are texture, color and shape, being the last one the most widely considered. For that reason, a symbol recognition system usually requires the definition of expressive and compact shape descriptors. The research on shape descriptors has been very intensive in last decades, and several surveys can be found in the literature [ZL04], [RTV07], [MKJ08].

It must be said that most of the symbol recognition methods are defined for presegmented symbols, because the recognition of non-segmented symbols is extremely difficult. For these reasons, some segmentation methods have been studied ([CW00], [Suw05]), and even some systems that perform segmentation and recognition in par-

allel, in other words, they use the recognition phase in order to supervise the resulting segmentation. As an example, a method for segmenting and recognizing two touching symbols is presented in [RRC07]. They evaluate several segmentation candidates produced by grouping elements of an over-segmentation.

The desirable properties of a shape-based approach for symbol recognition can be divided in two main groups, depending on the descriptor or the classifier. Concerning to the point of view of shape signature, the descriptor should ideally guarantee intra-class compactness and inter-class separability.

Symbol recognition descriptors can be classified depending on different taxonomies. Mehtre [MKL97] proposes a classification based on Boundary-based methods and Region-based methods. The first one only takes into account the shape boundary (contour) information, whereas the second one extracts information from the whole shape region. Contrary, Zhang [ZL04] proposes a division between global/statistical and structural methods. The first ones represent the image as a n-dimensional feature vector, whereas the second ones usually represent the image as a set of geometric primitives and relationships among them.

In this dissertation, we will use the Zhang's taxonomy but also classified depending on the kind of input symbol image: printed symbols, hand-drawn symbols and camera-based symbols (see Fig. 2.10). The main reason is that each kind of input image has different restrictions, requiring the solution to different problematics.

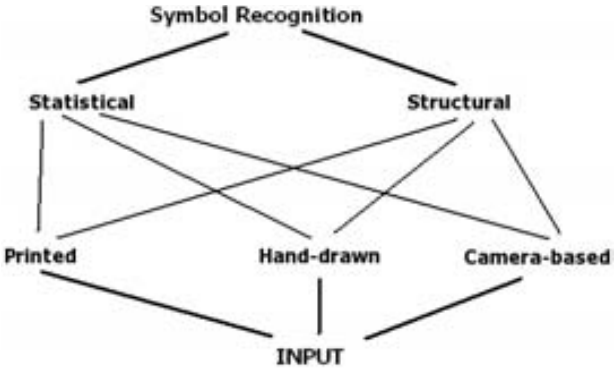


Figure 2.10: Classification of symbol recognition descriptors.

2.3.1 Recognition of Printed Symbols in Documents

The recognition of printed symbols has been a wide area or research in the last decades. It is key issue in the interpretation of engineering and architectural drawings, musical scores, maps, diagrams, etc., covering not only symbol recognition but also symbol indexing and spotting [TTD06],[FJ03].

A symbol recognition method should be tolerant to noise, degradation, occlusions and distortion (including shear), and due to isolated symbols which are present in graphical documents, it must also take into account the variations in rotation, scaling

and translation. According to Zhang [ZL04], symbol descriptors can be classified into statistical and structural approaches. These two groups will be commented next.

Statistical Symbol Recognition Methods

Statistical approaches tend to use pixels as the primitives to extract features from. They are usually few sensitive to noise, and domain independent. The main drawback is that the rotation and scale invariance is usually more difficult to achieve than structural methods, and in addition, it is difficult to add context information. Let us briefly review the most popular approaches:

Geometric features Some approaches [PPR00] use geometric features, such as centroids (center of gravity), area (number of active pixels), circularity (the ratio of the area of the shape to the area of a circle), bounding box (the smallest rectangle that contains the symbol), projection profiles (the distance of the contour pixels to the boundarys of the bounding box), axes of inertia (which represent the directions of the symbol) or zoning (after dividing the symbol in regions, it computes the area of each region) [DLDG$^+$00]. The main advantages are the simplicity and low complexity. The main disadvantages are that they are not rotation invariant and sensitive to noise and distortion.

Moment invariants Moment invariants are also applied to symbol recognition, being regular moments, Zernike moments and Legendre moments the most widely used. The main advantage is their capability for symbol reconstruction from features. **Regular moments** are not orthogonal, suffering from a high degree of information redundancy. **Zernike moments** [KH90] are moment-based descriptors, which are based on the Zernike polynomials and defined over the unitary disk on an orthogonal basis. They are rotation invariant and also quite robust to noise. They are slow to compute, thus there are some approaches for a fast computation of Zernike moments [GSTL02] or even real time computation [KA05]. **Legendre moments** are defined over the Legendre polynomials. They are also orthogonal, but are more severely affected by noise than the rest of the moments. A comparative study of moment invariants can be seen in [TC88].

Scale Space The main idea of Scale Space approaches are is that the original curve can be more and more simplified, and so small structures should vanish. The most representative is the **Curvature Scale Space** (CSS) [MM86], which was selected as a standard descriptor of the MPEG-7 [MSS02]. It is a contour approach, suitable for classifying symbols in data sets where the most important discriminant feature is the external contour (such as the MPEG-7 data set). It consists in smoothing the shape by convolving it with a Gaussian kernel. It is a robust contour-shaped descriptor, which captures the main features of the shape of the symbol. It is fast to compute and robust to changes in scale and rotation. The main disadvantages are that this descriptor can only be used for closed curves and it is sensible to noise.

Spatial relations Spatial interrelation features describe the region or the contour by the relation of their pixels or curves, such as Shape Context and Pixel-Level Constraint. One of the most commonly used in symbol recognition is the **Shape context** [BMP02]. Contrary to CSS, it can work with external and internal contour, and also with non-closed curves. It extracts a global histogram for each corresponding point, which are concatenated to form the context of the shape. The main drawback is that it requires point-to-point alignment of the symbols to be compared before their alignment. It is translation and scaling invariant, but its rotation invariance relies on the tangent at every pixel, which is unstable in presence of noise. **Pixel-Level Constraint** (PLC) [Yan05] is a contour-based descriptor, based on statistical integration of pixel-level constraint histograms. For each pixel, a histogram is constructed to figure out the distribution of constraints among other pixels. Then, a veature vector from these histograms is created. It is of a fixed size, so, alignment (like in Shape Context) is not necessary. It has quite good performance in front of deformed symbols, and it is rotation and scale invariant.

Image transforms Image transforms such as Fourier Transforms and their variants are classic techniques but still used in symbol recognition. The **Fourier descriptor** [KSK93] is obtained by applying the Fourier Transform on the shape boundary coordinates. It is rotation, translation and scale, although they have limited discriminatory power, and are sensible to noise. **General Fourier Transform** (GFT) [YNGR07] is based on the modified polar Fourier Transform, applies a 2D Fourier Transform to the polar representation of the image. The coefficients are conveniently normalized in order to achieve invariance to rotation and scale. **Fourier-Mellin Transform** [AOC$^+$00] computes the fourier transform in the angular parameter, whereas in the radial parameter is the Mellin-Transform, which is a kind of moment function in a complex variable. It is invariant to rotation and scale. **R-Signature** [TWS06] is based on Radon transform [TW02]. It is translation and scale invariant, and includes the 1-D Fourier transform to reach invariance to rotation. The Hough Transform has also been used for detecting symbols in line drawing images [FMKK00], thanks to its ability to detect lines using a voting scheme. **Ridgelets transforms** [TV06] are based on the application of wavelets [ABCdFC97] to the Radon transform [TW02] of an image. The ridgelets transform will detect singularities in the Radon space, combining advantages from both transforms, the ability to detect lines, from the Radon transform, and the multiscale property of wavelets to work at several levels of detail. They are rotation invariant (not to translation) and with a good performance in degraded symbols. **Angular Radial Transform** (ART) [KK99] decomposes the shape in an orthogonal basis, taking use of a radial and angular function. It has good perfomance for general shapes and uses few features by descriptor. Both ART and CSS are standards of the MPEG7 data set.

There are other descriptors that can not be included in the previous groups, such as the symbol descriptor based in **Kernel Density Estimation** presented in [ZLZ06]. Pre-segmented symbols are represented as 2D kernel densities from the shape skeleton, and for evaluating the similarity between symbols, the Kullback-Leibler divergence is used. It has very good performance in front of degraded and noisy symbols.

Structural Symbol Recognition Methods

In structural approaches, straight lines and arcs are usually the basic primitives for represent the shape. Strings, graphs or trees represent the geometrical and topological relations (perpendicularity, adjacency, parallel, crossing...) between these primitives. The similarity measure is therefore performed by string, tree or graph matching. The rotation and scale invariance is quite easy to achieve using structural descriptors, but they usually have a high computational cost. In addition, they are sensitive to noise and distortions in comparison to statistical approaches. They can be classified in several groups:

Graphs Graphs are a powerful shape representation; by assigning suitable semantic meaning to the vertices and edges, it can form a complete representation of the shape of the symbol. Usually, nodes correspond to points and lines of the image, and edges correspond to relations between these primitives. The matching consists in finding the best subgraph isomorphism between the two symbols to be compared. They allow segmentation and recognition at the same time. The main drawback is the high complexity cost. Moreover, they are sensitive to errors and noise. For this reason, error-tolerant subgraph isomorphism algorithms have been proposed [MB98]. **Region Adjacency Graphs** (RAG) with an error-tolerant subgraph isomorphism are proposed for symbol recognition in [LMV01]. In the approach proposed, regions are represented by polylines and string matching techniques are used to measure their similarity. The algorithm follows a branch and bound approach driven by the RAG edit operations, which reduces time complexity. The method shows good performance in front of distortions. **Attributed graph grammars** proposed by Bunke in [Bun82] are a combination of graphs and grammars. The advantage is that a grammar can store in a compact way all valid instances of a kind of symbols. The recognition consists in parsing its representation to test whether the symbol can be generated by the grammar. This approach can cope with partially occluded symbols, and are applied to domains in which the symbols can be defined by a set of rules, such as technical drawings, logic diagrams and flowcharts. **Structural signatures** proposed in [CGV+08] use topological graphs to describe the spatial organization of the segments. The graph is computed from segments, which are detected using Hough Transform. The classification is performed using a Galois Lattice classifier. It is robust to transformations, noise and degradation. The **Attributed Relational Graph** (ARG) is proposed in [SW08], which is based on arcs and segments. It is generated from the skeleton of the symbol, then, RANVEC is used for vectorizing it. The authors use genetic matching, because it is faster than graph matching. It is rotation, scale and translation invariant, but it is sensitive to segmentation errors.

Trees A tree is a simple version of graph, so, faster than graphs representation and manipulation. It discards some features to make the final representation more manageable. Spatial Division Tree and Directional Division Tree are two common techniques used for symbol recognition. The **Spatial Division Tree** (SDT) proposed in [LWJ04] consists in the following. For each node, a stroke of the drawing symbol is selected and the remaining strokes are divided into 2 groups: strokes in the left,

and strokes in the right side. The selection and division procedures are recursively repeated until there is only one stroke left in a node. It can be used for the recognition of incomplete graphic objects. The searching can be pruned heuristically, saving time and it is quite efficient. Contrary, the **Directional Division Tree** (DDT) proposed in [HZW08] generates the tree in a different manner. It selects a visually critical entity, and organize the rest entities according to their locations in 9 regions around the critical entity. The main drawback is that it is very important the decision of the critical entity. The main difference between SDT is that DDt performs division in a 2D way whereas SDT performs it in a 1D, becoming more complex.

Strings Strings are another frequently used structure for simplifying graphs, which provides a simple and compact representation. An example of a these techniques can be found in [TT89], where the **Attributed String matching** proposed is based on the representation of the contour as a Chain Code. The Chain Code [Fre61] is a contour-based representation which describes an object by a connected sequence of line segments with specified lengths and directions. The string matching method applies edit distance to compute the similarity between the chain code of two different shapes.

Others There are other structural approaches that can not be included in the previous groups. An example is the vector-based graphic symbol recognition system proposed in [YZL07], which is based on a mathematical model. The authors describe the geometric information of a primitive with respect to the whole symbols, and perform one-to-one matching from primitives of the test and the model shape. The approach is invariant to rotation and scale. Another example is the Network of Contraints. In [AST01] a **Network of Constraints** for recognizing architectural symbols is proposed, which is an adaptation of Messmer and Bunke's network approach [MB96]. The method is based on the description of the model through a set of constraints on geometrical features, and on propagating the features extracted from a drawing through the network of constraints. The main advantages are the possibility to incrementally build and update the model (adding new symbols), the adaptability and flexibility, and its independence of the geometry and topology of the symbols. The main drawback is that it requires pre-vectorized symbols, depending on the qualitiy of vectorization. **Deformable models** [VM00] are also used for describing pre-segmented symbols. The description of the symbol is based on a probabilistic model, consisting of a set of lines described by the mean and the variance of line parameters (midpoint position, orientation and length). It allows the automatic learning of shapes. They are invariant to distortions and rotation, but the basic primitives are lines, thus not being suitable for symbols with arcs and curves. **Hidden Markov Models** [CC01] can also be seen as structural methods, because the structure of the symbol can be described by the sequence of states, and the recognition consists in finding the sequence of states with higher probability. They are able to segment and recognize distorted symbols, although they are not usually rotation invariant.

2.3.2 Hand-drawn Symbol Recognition Methods

The particular case of hand-drawn symbols deserves an special attention. The main kinds of distortions in this case are: elastic deformation, inaccuracy on junctions or on the angle between strokes, ambiguity between line and arc, errors like over-tracing, overlapping, gaps or missing parts (see some examples of the typical distortions in hand-drawn symbols in Fig.1.8). Moreover, the system must cope with the variability produced by the different writer styles, with variations in sizes, intensities and the increase in the number of touching and broken symbols (see Fig.1.9). Some examples of hand-drawn symbols from architectural drawings and music scores are shown in Fig.2.11 . In this section, some of the techniques also applied to hand-drawn symbol recognition are commented.

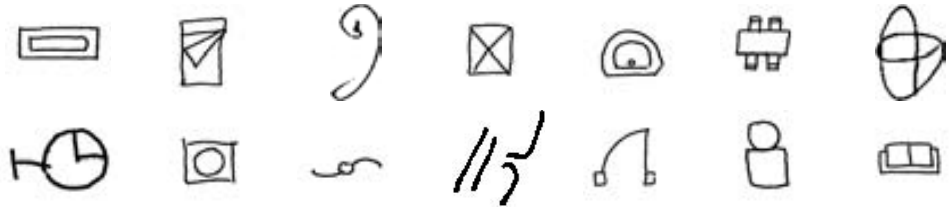


Figure 2.11: Examples of hand-drawn symbols.

Statistical Hand-drawn Symbol Recognition Methods

There are several approaches for hand-drawn symbol recognition in the literature. Zernike moments, Angular Radial Transform (ART) [KK99] and R-Signature [TWS06] are some examples of Region-based statistical approaches. Zernike moments are widely used for handwritten symbols (even online systems [HN04]), because they maintain properties of the shape, and are invariant to rotation, scale, and deformations. Concerning contour-based approaches, Curvature Scale Space [MM86] (CSS) has also shown quite good performance in case the data set is formed by hand-drawn symbols with closed curves. Shape Context [BMP02] has very good performance in hand-drawn symbols, because it is tolerant to deformations. These descriptors have been described in the previous section.

There are several **online statistical approaches** for symbol recognition, a few are briefly mentioned next: Hse and Newton [HN04] propose an online handwritten symbol recognition methods which uses Zernike moments; Parker et al. [PPR00] propose a method which is applied to pre-segmented symbols in logic diagrams, and uses geometric features and template matching. In the method proposed by Wilfong et al. [WSR96] the symbol is represented as a sequence of coordinates, and the matching is based in curvature distance. Miyao and Maruyama present in [MM07] a handwriting music symbol recognition system, consisting in the combination of two classifiers: the first one uses chain codes for representing the strokes, and string-edit distance is used for the matching; the second classifier is used for complex strokes, consisting in the division of the strokes into regions, and the computation of the

directional feature for each region. Afterwards, a trained Support Vector Machine (SVM) is used for the classification.

Structural Hand-drawn Symbol Recognition Methods

Some of the most common structural approaches for hand-drawn symbol recognition found in the literature are the Attributed Graph Grammars, Region Adjacency Graphs, Attributed Relational Graphs, Deformable Models and also Hidden Markov Models.

Attributed graph grammars proposed by Bunke in [Bun82] include a distortion model for allowing hand-drawn diagrams, which can also cope with partially occluded symbols. Region Adjacency Graphs [LMV01] are well-suited to describe symbols in hand-drawn architectural documents, showing good performance in front of distortions typically found in these documents. Deformable models [VM00] are also used in hand-drawn architectural documents. They are invariant to distortions and rotation, but the restriction is the use of lines (not curves) in the handwriting. Hidden Markov Models are also widely used in offline [MR00] and online symbol recognition methods [XCJW02], [ABS04].

Messmer and Bunke [MB96] propose a method for the recognition and the automatic learning of hand-drawn graphic symbols in engineering drawings. It allows model pre-compilation through the use of a network, where all model descriptions are gathered at once. The graphic symbols and the drawings are represented by attributed relational graphs. The recognition process is formulated as a search process for error-tolerant subgraph isomorphisms from the symbol graphs to the drawing graph.

Concerning **online structural approaches**, Fonseca et al. [FPJ02] propose a method for sketched architectural symbols, using fuzzy logic and geometric features; Peng et al. [PLWH04] propose a constrained partial permutation algorithm which uses binary and ternary topological spatial relationships for the recognition of symbols; and Mas et al. [MJSL08] describe a complete system for recognizing architectural drawings, representing the data in trees and proposing adjacency grammars with distortions measures for adapting them to sketches. Spectral models [LSL05] are also used for hand-drawn symbol recognition and retrieval. First, sketches are decomposed into basic geometric primitives and represented as a topological graph that encodes both the intrinsic attributes of the primitives and their relationships. The spectral graph descriptor is then adopted to translate the graph-match into the computation of vector distances. The method is invariant rotation and scale, and also to arbitrary drawing orders.

Finally, some approaches for recognizing mathematical symbols are also briefly mentioned. Mathematical symbol recognition requires a mixed strategy, because it requires text recognition and graphics (symbols) recognition. It is a very active research field (see [CY00] for a survey), which also includes several online systems: Shi et al. [SLS07] propose a symbol decoding and graph generation algorithm; and Garain and Chaudhuri [GC04] develop a full mathematical expression recognizer system, which involves symbol recognition (using both online and offline features) and structural analysis of multistroke characters using context free grammars (CFG).

2.3.3 Camera-based Symbol Recognition Methods

Camera-based symbol recognition is an emerging area of study in which symbols are detected from real images using cameras. In this type of applications the system should cope with a totally different problematic: uncontrolled environments, illumination changes, and changes in the point of view (perspective). Some examples of symbols that can appear in real images are shown in Fig. 2.12.

Figure 2.12: Examples of symbols in real environments.

In the cases where symbols are detected from real images, the SIFT descriptors and its variants are the strategies most frequently applied [Low04]. The SIFT descriptor is based on determining the significant orientations within a region taking into account their spatial arrangement.

Other approaches include the combination of one method for detecting symbols in real images (such as the Stein and Medioni's method [SM92] for computing the relevant features) and symbol descriptor methods. In [SG07] a color text recognition method is proposed, which uses a convolutional neural network architecture. Sun et al. propose in [SWW07] a printed digit localization and recognition approach, which uses connected components for detection and an artificial neural network. In [MTM07] a engraved character recognition method is proposed, using a classifier based on geometrical features and a multi-layer perceptron. In [RLD07] vector signatures are used for symbol detection. A vector signature is defined based on accumulated length and angular information computed from polygonal approximation of contours. It is not necessary to perform a previous segmentation. It is invariant to rotation, scale, translation, distortions, slight changes in perspective and blurring.

2.3.4 Conclusion

In this section we have reviewed the main symbol recognition methods, specially for hand-drawn symbols. Table 2.3 shows a summary of the behavior of some of the most common approaches described in this section. One can see that the robustness to affine transformations is achieved in all the methods, but some of them (e.g. Zernike and ART-based approaches) are more sensible to noise (degradation) than others. Concerning the typical distortions appearing in hand-drawn symbols, some methods have shown very good performance (e.g. RAG and Shape Context matching). It is important to remark that some methods (e.g. Zernike moments-based

approach, Shape Context matching) are more suitable than others for describing complex symbols, such as the symbols composed of several curvilinear strokes (e.g. the music treble clef). Finally, some methods can be used both for offline and online symbol recognition (e.g. Zernike moments-based approach), whereas others, are more suitable for offline recognition.

Reference	Method	Taxonomy	Affine Transf.	Noise	Hw. Distort.	Primitives	Input On./Off.
[BMP02]	Shape Context	Region, Cont.	Rotation, Scale	Yes	Yes	Complex	Offline
[MM86]	CSS	Silhouette, Cont.	Rotation, Scale	Yes	Partial	Complex	Offline
[HN04]	Zernike Moments	Region, Cont.	Rotation, Scale	Partial	No	Complex	Both
[KK99]	ART	Region, Cont.	Rotation, Scale	Partial	No	Complex	Offline
[TWS06]	R Signature	Region, Cont.	Rotation, Scale	Yes	Partial	Basic	Offline
[Low04]	SIFT	Region, Cont.	Rotation, Scale	Yes	No	Complex	Offline
[LSL05]	Spectral Graphs	Region, Struct.	Rotation, Scale	Yes	Yes	Basic	Online
[Bun82]	Attr. Graph Grammars	Region, Struct.	Rotation, Scale	Yes	Yes	Basic	Offline
[LMV01]	RAG	Region, Struct.	Rotation, Scale	Yes	Yes	Basic	Offline
[VM00]	Deformable Models	Region, Struct.	Rotation, Scale	Yes	Yes	Basic	Both

Table 2.3: Symbols Recognition methods: Approach considered, Taxonomy (Region/Silhouette, Continuous/Structural), Robustness to Affine transformations, Noise, Typical Distortions in hand-drawn symbols (such as elastic deformations), Symbols (Basic/Complex), and finally, kind of input image (Online/Offline).

Due to the large different kinds of problems found in symbol recognition applications, some approaches are better than others depending on the application field, and it is very difficult to find a symbol recognition method that suits in most fields, outstanding the rest of approaches. In fact, most methods are defined for coping with a specific problematic, obtaining very high results in this problem, but reaching discrete results when coping with other problems. For this reason, more research must be done in this field.

Concretely, symbol recognition methods for hand-drawn symbols must cope with the high variability in the visual symbols' shape, which has been produced by the different writer styles. Since the distortion is an important problem in this field, rougher descriptors should be researched, for obtaining the general structure of the symbol, and avoiding the confusing style variations.

Chapter 3

DTW-based Hand-drawn Symbol Recognition method

The first approach for writer identification in old music scores will use symbol recognition methods. One of the major difficulties of handwriting symbol recognition is the high variability among symbols because of the different writer styles. In this chapter we introduce a robust approach for describing and recognizing hand drawn symbols tolerant to these writer style differences. This method, which is invariant to scale and rotation, is based on the Dynamic Time Warping (DTW) algorithm. The symbols are described by vector sequences, a variation of the DTW-distance is used for computing the matching distance, and K-Nearest Neighbor (k-NN) is used to classify them. Our approach has been evaluated in two benchmarking scenarios consisting of hand drawn symbols. Compared with state-of-the-art methods for symbol recognition, our method shows higher tolerance to the irregular deformations induced by hand drawn strokes.

3.1 Introduction

The first writer identification approach will be based on symbol recognition. For this purpose, three symbol recognition approaches have been proposed, dealing with the high variability among symbols. The first symbol recognition method is described in this Chapter, whereas the other two approaches are described in next Chapter.

Hand-drawn symbol recognition is a particular case of handwriting recognition, which is one of the most significant topics within the field of Document Image Analysis and Recognition (DIAR). Over the last years, relevant research achievements have been attained. Simultaneously, commercial products have become available. The progress has been noticeable in applications like bank check processing, postal sorting, historical document transcription or on-line recognition in calligraphic interfaces. A parallel use has also been explored in writer identification for forensic sciences and writer verification in signatures. Handwriting recognition is a difficult problem due to the variability among scripts and writer styles, or even between different time periods.

Due to that, commercial applications are usually constrained to controlled domains that make use of contextual or grammatical models and dictionaries. The type of source data (handwritten separate characters vs cursive script) is also an important constraint. Focusing on cursive script recognition, the recognition approaches can roughly be classified into *analytical* or *holistic* methods. Analytical methods perform a segmentation preprocess that divides the word image in sequences of smaller units which are therefore classified in terms of associated features and lexical information. Holistic methods, which recognize words as a whole, usually describe the word image as a unidimensional signal consisting of a sequence of image features at each column. This allows to use techniques sometimes inspired by the speech recognition domain such as sequence alignment by dynamic programming [KL83] or Hidden Markov Models [Rab89].

Although the analysis of textual handwritten documents has an intensive activity, the analysis of hand-drawn documents with graphical alphabets is an emerging subfield. Due to the fact that architectural, cartographic and musical documents use their own alphabets of symbols (corresponding to the domain-dependent graphic notations used in these documents), the graphics recognition community has developed specific methods for understanding graphical alphabets. These techniques are different from the classical methods used for Cursive Script Recognition. Two major differences between the two problems can be stated. Cursive script recognition has the context information in one dimensional way, but graphical alphabets usually are bidimensional. In addition, the use of syntactical knowledge, and lexicons, is more effective in text recognition than in diagrammatic notations because of the variability of structures and alphabets of the latter.

As it has been commented in Chapter 1, the particular case of hand-drawn symbols deserves a special attention. A hand-drawn symbol recognition method must cope with elastic deformations, inaccuracy on junctions or on the angle between strokes, ambiguity between line and arc, errors like over-tracing, overlapping, gaps, missing parts (see Fig. 1.8), and the variability produced by the different writer styles (see Fig. 1.9). Techniques used in the classification of handwritten shapes have been analyzed and verified in a specific domain: the recognition and classification of symbols with high variability (i.e. musical clefs) due to the different writer styles. In this case, we can affirm that there is no clear separability between classes using the common hand-drawn descriptors. The main reason is that the huge variability present in these symbols confuses the system, because symbols belonging to the same class are very different (distortions are very important), so the descriptor obtains quite different descriptions. In a similar way, symbols that belong to different classes can be very similar, so their descriptors are also similar. For that reason, other descriptors for hand-drawn symbols are required, specially rougher descriptors, which can obtain the general structure of the symbol, avoiding confusing details.

In this Chapter we propose a method inspired by the holistic approaches for unconstrained handwritten word recognition, but extended to bidimensional shapes appearing in bidimensional layouts. The proposed method is robust against the elastic deformations typically found in handwriting and invariant to rotation and scale. The method proposed is based in the Dynamic Time Warping (DTW) algorithm [KL83] for signals (one-dimensional data) and it has been extended to graphical symbols

(two-dimensional data). Among the two major families of methods for handwritting recognition, namely sequence alignment (e.g. DTW) and Hidden Markov Models (HMMs), our work is based on the former. The DTW algorithm has been successfully used for finding the best match between two time series in a noisy and complex domain. It has been already used in handwritten text recognition [RM03], coping with the elastic deformations and distortions in the writing style. For that reason, we maintain that the DTW algorithm can be adapted for the recognition of hand drawn symbols. In comparison to HMMs, the DTW approaches are more suitable for coping with the problem of hand drawn symbol recognition when there is a low number of instances for each symbol (which is the case of some hand drawn graphical databases), not being enough for a successful training process. In addition, the adaptation of DTW to a rotation-invariant system is easier than the adaptation of HMM because HMM requires to train a model for each possible orientation, with the consequently increment of its time complexity.

To solve the problem of rotational invariance, classical and effective methods exist in the literature on OCR or Symbol Recognition. Methods like projections in different orientations or zoning using concentric ring masks are well-known. We have taken into account these ideas and extended them to a novel DTW-based algorithm. The steps of the method proposed are the following. First, column sequences of feature vectors from different orientations of the two input shapes to be compared are computed. The features comprise the upper and the lower profile and the number of pixels per region. Once we have the features for all the considered orientations, the DTW algorithm computes the matching cost between every orientation of the two symbols, and decides in which orientation these two symbols match with the lowest cost.

The rest of the Chapter is organized as follows. In Section 2 the fundamentals of the Dynamic Time Warping (DTW) algorithm are presented. Afterwards, our DTW-based method for the recognition and classification of graphical symbols is fully described in Section 3, demonstrating its invariance to rotation and scale. In Section 4, the experimental results are presented. Finally, concluding remarks are reported in Section 5.

3.2 Fundamentals of the Dynamic Time Warping

Since the approach proposed is based in the Dynamic Time Warping (DTW) algorithm, we will start with a short review of the state of the art and the background of DTW before detailing our approach.

3.2.1 State of the Art of DTW

The Dynamic Time Warping algorithm (DTW) was first introduced by Kruskal and Liberman [KL83] for putting signals into correspondence. It is a much more robust distance measure for time series than Euclidean distance, allowing similar samples to match even if they are out of phase in the time axis. DTW can distort (or warp) the time axis, compressing it at some places and expanding it at others, finding the best matching between two samples. This technique was first used in the context of

speech recognition, a domain in which the time series are notoriously complex and noisy. The method was used for coping with noise and variations in speech speed.

This technique has been also used in audio analysis. In [HDT03] DTW is used for aligning polyphonic audio recordings of music to symbolic score information in standard MIDI files without any polyphonic transcription. Also, Schwarz et al. [OS01] and [SRS03] propose a methodology based in DTW for the automatic alignment of music recordings, where the spectral peak structure is used to compute the local distance, enhanced by a model of attacks and of silences. The authors say that it is able to cope with polyphonic music, multi-instrument music, vibrato, fast sequences, and it is even useful as an indicator of interpretation errors.

Besides audio and speech recognition [RJ93], DTW has been widely used in many other applications: In chemical engineering, it was used for the synchronization and monitoring of batch processes in polymerization [GP95]. DTW has been successfully used in gesture recognition to align biometric data [GD95], signatures [MP99], fingerprints [KV00] and even for managing constant image brightness [CRH95] (matching two intensity histograms). Many researchers have demonstrated the utility of DTW for ECG pattern matching [CPB+98]; while in robotics, Schmill et al. demonstrated a technique that utilizes DTW to cluster an agent's sensory outputs [SOC99]. DTW was also introduced into the Data Mining community ([KP99], [RK05]) as an utility for various tasks for time series problems including classification, clustering, and anomaly detection. In this field, Keogh and Ratanamahatana have defined an exact indexing of DTW [KR05].

In bioinformatics [AC01],[CSE03],[CS06], DTW has been successfully applied to genomic expression data. It must be said that both implementations introduce a time-symmetric version of DTW, modifying the original DTW algorithm. The authors claim that their version is more efficient and simpler and yields the same time warp distance when computed left to right as from right to left. This feature allows an unambiguous computation of the Boltzmann probability that two time points are aligned in an optimal time warping of genes.

An interesting work done by Oates et al. is the combination of DTW with Hidden Markov Models (HMM) [OFC99]. Concretely, they present a hybrid time series clustering algorithm that uses DTW and HMM induction. The two methods complement each other: DTW produces a rough initial clustering (estimating the number of generating HMMs) and the HMM iteratively removes from these clusters the sequences that do not belong to them. Finally, the process converges to a final clustering of the data and a generative model for each of the clusters.

Finally, Rath and Manmatha have applied DTW to the handwritten recognition field [RM03], [RM02], coping also with the indexation of repositories of handwritten historical documents. Also Manmatha [KMA04] proposed an algorithm based on DTW for a word by word alignment of handwritten documents with their (ASCII) transcripts. DTW has been also used for recognizing greek characters [VGPS07]. The method computes a vector of about 300 features, consisting in horizontal and vertical density zones (zoning), projections of the upper/lower/left/right profiles, distances from character boundaries, profiles from the character edges. Afterwards, they perform dimensionality reduction, and apply the DTW algorithm.

Concerning online handwritting recognition, some work has also been done. It

has been applied for online tamil and tegulu scripts recognition [PBS+07]. The authors compare the use of different features, such as the x, y coordinates, their derivatives and curvature features; shape context and tangent angle, and the generalised shape context. Then, they use DTW and k-NN for the final classification. Vuori et al. [Vuo02],[VLOK01],[VLOK00] have developed an online handwriting recognition system, which is able to recognize handwritten characters of several different writing styles and improve its performance by adapting itself to new writing styles. The recognition system is based on prototype matching using DTW. The classifier is based on the k-NN rule and it is adapted to a new writing style by adding new prototypes, inactivating confusing prototypes, and reshaping existing prototypes using a Learning Vector Quantization (LVQ)-based algorithm. Finally, Niels presents in [Nie04] and [NV05] some modifications of the DTW technique described by Vuori et al., adding new constraints and two different averaging techniques for merging members from the same cluster into a single prototype. Concretely, his goal is to retrieve a set of best matching allographic prototypes (an allograph is a variant shape of a letter or phoneme) based on a query input character from an online handwriting system.

3.2.2 Background of the DTW

In this part of the section, the original DTW algorithm for signals is described. Afterwards, some approaches for adapting the DTW for two dimensional shapes are briefly discussed.

DTW for 1-Dimensional Signals

The DTW algorithm [KL83] is used for comparing signals by matching two one-dimensional vectors. It is a much more robust distance measure for time series than Euclidean distance, allowing similar samples to match even if they are out of phase in the time axis (see Fig. 3.1). DTW can distort (or warp) the time axis, compressing it at some places and expanding it at others, finding the best matching between two samples.

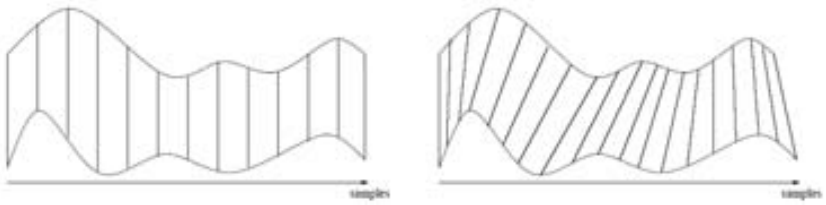


Figure 3.1: Normal and DTW alignment, extracted from [RM03].

Let us define the DTW distance of two time series $C = x_1..x_M$ and $Q = y_1..y_N$ in terms of the cost function $DTWCost(C, Q)$ (see Fig. 3.2(a)). For this purpose, a matrix $D(i, j)$ (where $i = 1..M, j = 1..N$) of distances is computed using dynamic programming:

$$D(i,j) = min \left\{ \begin{array}{l} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{array} \right\} + d2(x_i, y_j) \tag{3.1}$$

$$d2(x_i, y_j) = x_i - y_j \tag{3.2}$$

Performing backtracking along the minimum cost index pairs (i,j) starting from (M,N) yields the warping path (Fig. 3.2(b)). Finally, the matching cost is normalized by the length Z of this warping path, otherwise longest time series should have a higher matching cost than shorter ones. Therefore, the cost function is defined as follows:

$$DTWCost(C,Q) = D(M,N)/Z \tag{3.3}$$

The creation of this path is the most important part of their comparison: it determines which points match (Fig. 3.2(c)) and are to be used to calculate the distance between the time series. In addition, DTW is able to handle samples of unequal length, allowing the comparison without resampling.

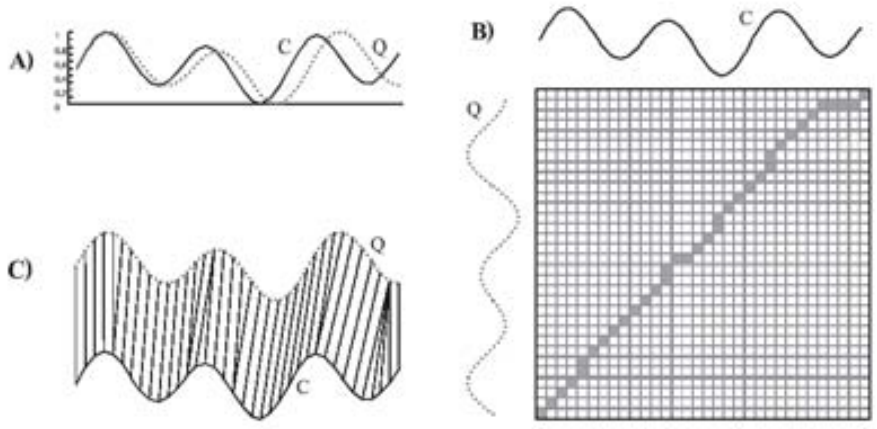


Figure 3.2: An example of DTW alignment (extracted from [KR05]) a) Samples C and Q. b) The matrix D with the optimal warping path in grey color. c) The resulting alignment.

DTW for 2-Dimensional Shapes

In case of bidimensional data, the DTW computation must be adapted. Some work has been done in the adaptation of DTW to 2 dimensions (see [LP92],[US98]), but these approaches are of a very high time complexity, reaching $O(N^{4N})$ and $O(N^3 9^N)$ respectively. For this reason, some research work has been focused on the reduction of the 2D problem. Generally, the reduction of dimensionality can be performed when 2D data can be encoded by 1D signals, such as shapes described by their external

contours (silhouettes). Specifically, for handwritten text methods, the 2D representation is typically reduced to 1D based on the assumption that text follows a given text line [RM03]. In these cases, the DTW computation can be easily applied, reducing significantly the time complexity of the 2D-DTW computation.

3.3 A DTW-Based Approach for Graphical Symbol Recognition

The basic dynamic time warping algorithm achieves good results when working with one-dimensional data and with handwritten words in documents. Concerning the hand drawn symbol domain, the method must be adapted to cope with the variations in writing style and rotation. In this section, the architecture for our DTW-based system is fully described and its benefits for hand drawn symbol recognition are presented. Comparing to the classical DTW, the proposed method introduces two main changes: first, different features are used and second, the computation of the DTW distance has been modified, combining information at certain orientations of the symbol.

3.3.1 Extraction of Features

The choice of features that better represent shapes is a key decision of the application of the DTW algorithm. In this work we have been inspired by features representing series with a view to reduce the dimensionality. Let us first describe the approaches of Rath and Manmatha [RM03], Marti and Bunke [MB01] and Vinciarelli et al. [VBB04] on which we have inspired our proposed representation.

Rath and Manmatha In the handwritten text recognition system described by Rath and Manmatha [RM03], the following four features are computed for every column of a word image: the number of foreground pixels in every column; the upper profile (the distance of the upper pixel in the column to the upper boundary of the word's bounding box); the lower profile (the distance of the lower pixel in the column to the lower boundary of the word's bounding box); and the number of transitions from background to foreground and viceversa. In this way, two word images A and B can be easily compared using DTW. If $f_k(a_i)$ corresponds to the k-th feature of the column i of the image A, and $f_k(b_j)$ corresponds to the k-th feature of the column j of the image B, the matching distance $DTWCost(A, B)$ is calculated using the same equations (eq. 3.1, 3.3) as in Kruskal's method, but instead of the equation 3.2, the computation of $d2$ will be the sum of the squares of the differences between individual features:

$$d2(x_i, y_j) = \sum_{k=1}^{4} (f_k(a_i) - f_k(b_j))^2 \tag{3.4}$$

Marti and Bunke Another typical set of column features in the literature is the one proposed by Marti and Bunke [MB01] for handwritten word recognition. The

following nine features are obtained per column: the number of foreground pixels, the center of gravity, the second moment order, the lower and upper profile, the differences between the lower and upper values with respect to the previous column, the number of gaps, and the number of pixels between the upper line and baseline of the word.

Vinciarelli et al. Finally, the features described by Vinciarelli et al. [VBB04] are also very common in the literature, consisting in a sliding window which moves from left to right. In this case, instead of the single column features, the window comprises several columns. After adjusting the size of the window to the area which contains pixels, it is divided into a 4x4 cell grid, and the number of pixels in every cell is used as a feature. Finally, the 4x4=16 features are concatenated to a 16-dimensional feature vector.

Our proposal Inspired by the approaches presented above, we propose a novel set of features for symbol description. In this field, it is important to obtain some information about the external shape (profiles), but also about the internal shape (distribution of pixels inside the silhouette). For this reason, in addition to the upper and the lower profile, our method divides every column in several regions, counting the number of foreground pixels per region (it can be seen as a column zoning). First, the image is normalized in terms of its size, and the following features are computed for every column of the image:

- f_1 is the upper profile.
- f_2 is the lower profile.
- $f_3 ... f_S$ are the number of foreground pixels in every region.

When computing the upper and lower profile, a morphological closing operation over the image is performed, so that few little gaps in the writing will not affect the final profile. Finally, all the features are normalized ($0 \leq f_k \leq 1$, k=1..S) and the features corresponding to the sum of pixels ($f_3, \ldots, f_S$) are smoothed over the symbol's columns using a gaussian filter for a better matching. Notice that due to the high variability in the writing style, the number of transitions per column (from background to foreground and viceversa) can confuse the system, thus, they are not used as features.

Figure 3.3 shows an example of the features extracted for the marked column of a music symbol: the pixels of the column are used for extracting the upper and the lower profile. Then, the column is divided in three equal regions (in this example, S=5), and for every region the number of pixels is counted.

The reader should notice that the features $f_3, ..., f_s$ provide an adequate information about the distribution of the pixels inside the shape. The number of regions is a parameter that can be set up to reflect the complexity of the symbols in the database. These measures will help to classify correctly shapes that have the same external contour but differences in their inner part. Moreover, it will not get confused when comparing axially symmetrical symbols. In Figure 3.4(b) one can see two similar images in terms of silhouette (both are squares), but very different inside (a

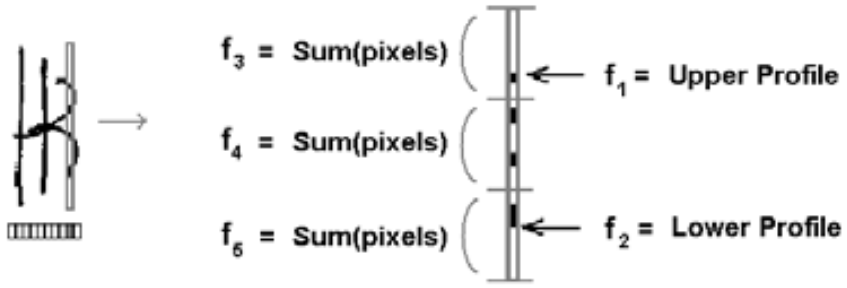


Figure 3.3: Example of features extracted from every column of the image, with $s = 5$: f_1 = upper profile, f_2 = lower profile, $f_3..f_5$ = sum of pixels of the image of the three regions defined.

cross or a circle). Notice that the upper/lower profiles and the whole sum of pixels per column are very similar (see fig. 3.4(a)), whereas the functions of the sum of the three regions (see Fig. 3.4(c)) are very different, being able to discriminate these two symbols.

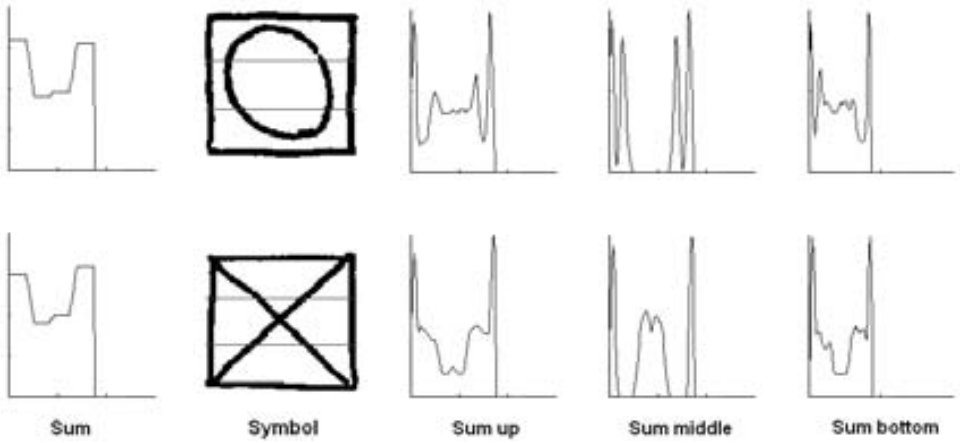


Figure 3.4: Two architectural symbols with similar external contour (squares) but with differences inside the contours (circle and cross). The first row corresponds to the features for the square with a circle, and the second row corresponds to the features for the square with a cross. a) Functions of the sum of pixels per column. b) Symbols. The grey horizontal lines divide the image in three regions: upper, lower and middle c) Functions corresponding to the sum of pixels for the upper, middle and bottom region. Notice that the functions in (a) are similar whereas functions in (c) are very different.

3.3.2 Computation of the DTW Distance

Due to the fact that the slant and the orientation of graphical symbols are frequently different between each other (see Fig.3.5), symbols can not be directly and easily compared between them. To cope with rotation invariance and hand drawn distortion, we define a DTW-based distance in terms of different projections, covering the full range of possible orientations of the symbol.

(a) (b)

Figure 3.5: a) Clefs: Two treble clefs with different slants. b) Two identical architectural symbols but in different orientations.

Let us introduce the notation that will be used in this section:

- A_α: Symbol A oriented at α degrees.
- B_β: Symbol B oriented at β degrees.
- $\mathtt{a}_{\alpha_i}$: Column i of the symbol A oriented at α degrees.
- $\mathtt{b}_{\beta_j}$: Column j of the symbol B oriented at β degrees.
- $D_{\alpha,\beta}(i,j)$: Matrix which contains the cost of matching the first i columns of A_α and the first j columns of B_β.
- $MC(\alpha,\beta)$: Matrix which contains at the position (α,β) the matching cost between A_α and B_β.
- $G(\alpha,\beta)$: Matrix which contains at the position (α,β) the sum of $MC(\alpha,\beta)$ and $MC(\alpha+90,\beta+90)$.

There are three steps in the procedure: the extraction of features at different orientations; the computation of the matching distance between all the possible combinations of orientations between the two symbols; and the computation of the final matching cost. In the first step, the two symbols A and B are oriented in certain angles (see Fig.3.6(a)), covering the range from 0 to 180 degrees. For each orientation, the column sequence of feature vectors (see Fig.3.6(b)) defined in the previous section is obtained. In the second step, the DTW distance is computed for every combination of orientations of the two symbols. Thus, every orientation of the symbol A is compared to every orientation of the symbol B. It should be observed that it is necessary to obtain the features from every orientation of the two symbols, because we do no know a priori which orientation will give the highest discriminatory power. Finally,

the third step consists in determining the final matching cost, and the two angle orientations in which the two symbols match with the lowest cost. In fact, we can not trust in only one matching when working with 2D data, because false matchings could appear if only one direction is used (see Fig. 3.7). For this reason we also take into account the perpendicular alignment in respect to the orientation we are considering. As a summary, we can define the final matching cost $DTWCost_{A,B}$ of the symbol A and B as the minimum of the results of summing $MC(\alpha, \beta) + MC(\alpha + 90, \beta + 90)$ for each possible α, β angles. These steps are fully described next.

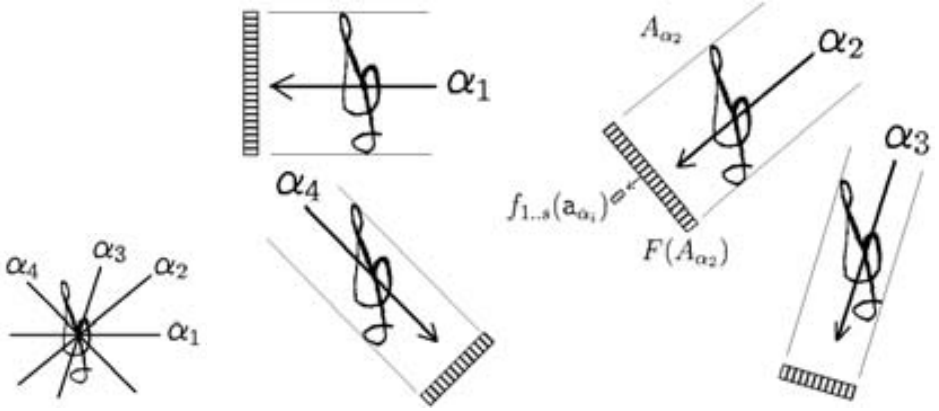


Figure 3.6: Example of feature extraction. (a) Some of the orientations used for extracting the features of every symbol. (b) Feature vectors extracted from every orientation $(\alpha_1, ...\alpha_4)$.

Extraction of Features Let us denote as $A_\alpha = (\mathtt{a}_{\alpha_1}, \mathtt{a}_{\alpha_2}, ..., \mathtt{a}_{\alpha_M})$ the symbol A oriented at α degrees, and $B_\beta = (\mathtt{b}_{\beta_1}, \mathtt{b}_{\beta_2}, ..., \mathtt{b}_{\beta_N})$ the symbol B oriented at β degrees. First, the column sequences of feature vectors $F(A_\alpha)$ and $F(B_\beta)$ are computed as it has been explained in the above section (the upper/lower profile and the sum of pixels per region):

$$F(A_\alpha) = \begin{pmatrix} f_1(\mathtt{a}_{\alpha_1}) & f_1(\mathtt{a}_{\alpha_2}) & ... & f_1(\mathtt{a}_{\alpha_M}) \\ f_2(\mathtt{a}_{\alpha_1}) & f_2(\mathtt{a}_{\alpha_2}) & ... & f_2(\mathtt{a}_{\alpha_M}) \\ ... & ... & ... & ... \\ f_s(\mathtt{a}_{\alpha_1}) & f_s(\mathtt{a}_{\alpha_2}) & ... & f_s(\mathtt{a}_{\alpha_M}) \end{pmatrix} \tag{3.5}$$

$$F(B_\beta) = \begin{pmatrix} f_1(\mathtt{b}_{\beta_1}) & f_1(\mathtt{b}_{\beta_2}) & ... & f_1(\mathtt{b}_{\beta_N}) \\ f_2(\mathtt{b}_{\beta_1}) & f_2(\mathtt{b}_{\beta_2}) & ... & f_2(\mathtt{b}_{\beta_N}) \\ ... & ... & ... & ... \\ f_s(\mathtt{b}_{\beta_1}) & f_s(\mathtt{b}_{\beta_2}) & ... & f_s(\mathtt{b}_{\beta_N}) \end{pmatrix} \tag{3.6}$$

Notice that the length of every column sequence of feature vector depends on the number of columns (the width) of the projection, and varies from one orientation to another.

Computation of the matching distance Once the column sequences of feature vectors are computed, the matching cost $MC(A_\alpha, B_\beta)$ between them must be calculated. First, the matrix D will be filled in with the classical DTW method:

$$D_{\alpha,\beta}(i,j) = min \left\{ \begin{array}{l} D_{\alpha,\beta}(i,j-1) \\ D_{\alpha,\beta}(i-1,j) \\ D_{\alpha,\beta}(i-1,j-1) \end{array} \right\} + d2(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) \tag{3.7}$$

The way of computing the distance *d2* must take into account that both the upper/lower profile features and the set of sum of pixels features have to be weighted equally in the calculation. The goal is to avoid a reduced effect of the upper/lower profile in the computation of *d2* whenever the feature number S is very high (which means a high number of regions for the zonning) For this reason, the two parts are weighted by 0.5 in equation 3.8:

$$d2(\mathtt{a}_{\alpha_i}, \mathtt{b}_{\beta_j}) = 0.5(\sum_{k=1}^{2}(f_k(\mathtt{a}_{\alpha_i}) - f_k(\mathtt{b}_{\beta_j}))^2 + 0.5(\sum_{k=3}^{s}(f_k(\mathtt{a}_{\alpha_i}) - f_k(\mathtt{b}_{\beta_j}))^2 \tag{3.8}$$

Then, the matching cost of A_α and B_β is normalized by the length Z of the warping path (obtained performing backtracking on $D_{\alpha,\beta}$), and this value is stored in the corresponding cell of the matrix MC:

$$MC(\alpha,\beta) = D_{\alpha,\beta}(M,N)/Z \tag{3.9}$$

This process must be repeated for all the orientations $\alpha = 1$ .. 180 and $\beta = 1$ .. 180 (the step is decided ad-hoc), filling all the cells in the matrix *MC*. Thus, every cell of the matrix $MC(\alpha,\beta)$ will contain the matching cost between the two symbols, the first one with an orientation angle of α degrees, and the second one with an orientation angle of β degrees. This means that if the two symbols are oriented in W different angles, the DTW distance is computed W^2 times.

Computation of the Final Matching Cost The next step is the computation of the final matching cost. It must be noticed that defining the final matching cost as the minimum of the DTW distances computed is not a good solution. For example, two symbols, which belong to different classes, could reach the minimum matching cost if they are oriented in some specific α and β angles, but they could have very high matching costs in other orientation angles. One way to avoid this problem is to look at the perpendicular alignment in respect to the orientation we are examining. Another option could be to have into account the matching cost of all the alignments, but it has been experimentally shown that it does not increment the discriminatory power whereas the time complexity is increased. As an example of the problem of using only one matching, Figure 3.7 shows the feature vectors of two different music symbols: in Fig.3.7(a) one can see that despite the two symbols being extremely different, only the upper contour and the middle sum are adequately different functions in the DTW sense, whereas in Fig.3.7(b) all the five functions of the first symbol are very different from the ones of the second symbol. For this reason, we should claim that

two symbols are correctly matched in α and β orientation angles, only if they have a low matching cost in α and β angles but also a low matching cost in the corresponding perpendicular alignment ($\alpha + 90$ and $\beta + 90$ degrees). For this step, let's define as G the matrix which stores in position (α, β) the cell $MC(\alpha, \beta)$ plus its corresponding perpendicular angle:

$$G(\alpha, \beta) = MC(\alpha, \beta) + MC(\alpha + 90, \beta + 90) \tag{3.10}$$

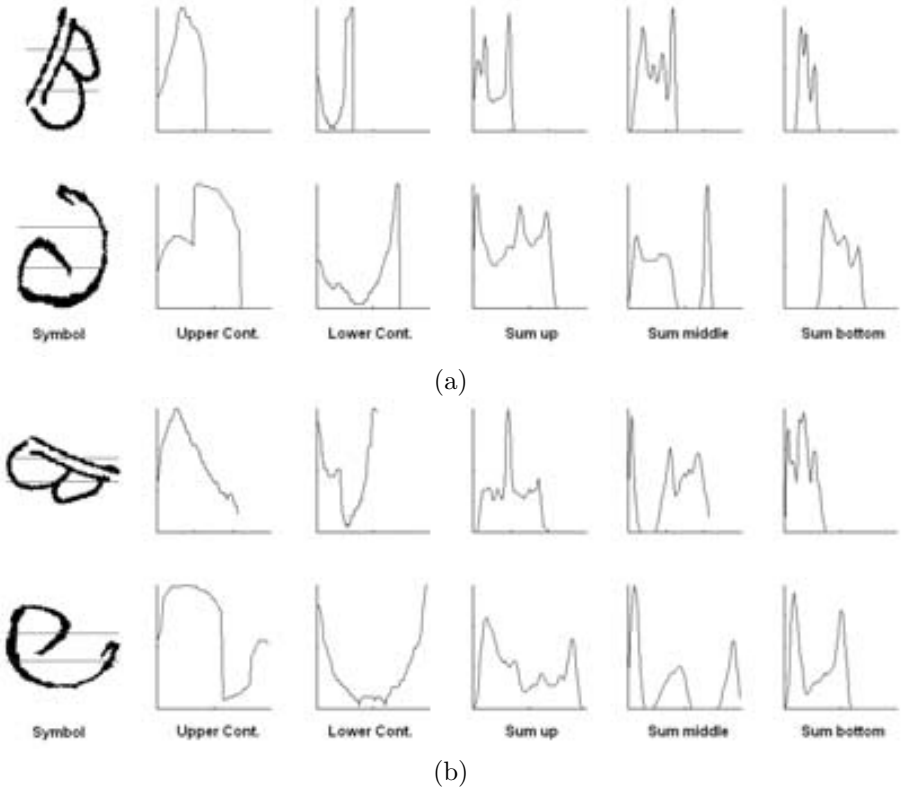


Figure 3.7: Feature vectors of two different music symbols: (a) The first symbol is an alto clef with a orientation of α degrees, the second one is a bass clef with a orientation of β degrees. b) The same alto clef with a orientation of $\alpha + 90$ degrees and the bass clef with a orientation of $\beta + 90$ degrees. Here the functions of the two symbols are very different.

Thus, the matching cost $DTWCost_{A,B}$ of the symbols A and B will be defined as the minimum value of the matrix G, where the angles θ and λ correspond to the orientation angles in which the two symbols are matched:

$$DTWCost_{A,B} = min(G) \tag{3.11}$$

The pseudo-code of the algorithm is given in Algorithm 1.

Algorithm 1 Dynamic Time Warping-based algorithm.

Require: Two symbols A and B
Ensure: The matching cost $DTWCost_{A,B}$
1: Obtain $F(A_\alpha), \forall\alpha \in \{0...180\}$.
2: Obtain $F(B_\beta), \forall\beta \in \{0...180\}$.
3: Compute the matching cost matrix MC as follows:
4: **for** each angle $\alpha \in \{0...180\}$ **do**
5: **for** each angle $\beta \in \{0...180\}$ **do**
6: Compute $MC(\alpha, \beta)$
7: **end for**
8: **end for**
9: Add the matching cost of every angle+90 degrees as follows:
10: **for** each angle $\alpha \in \{0...180\}$ **do**
11: **for** each angle $\beta \in \{0...180\}$ **do**
12: $G(\alpha, \beta) = MC(\alpha, \beta) + MC(\alpha + 90, \beta + 90)$
13: **end for**
14: **end for**
15: $DTWCost_{A,B} = min(G)$

Finally, it must be noted that with the proposed descriptor and matching strategy we obtain a symbol descriptor and classifier methodology which is rotation invariant and robust against typical elastic deformations present in hand drawn symbols. Concerning the complexity of the algorithm, if W corresponds to the number of angles in which every symbol is oriented, and N is the number of columns of the widest symbol image, then the complexity is $O(W^2N^2)$, because the DTW matching distance with order $O(N^2)$ is computed W^2 times. In the worst case, $W = N$, and the complexity is $O(N^4)$. This complexity cost is remarkably lower than $O(N^{4N})$ and $O(N^3 9^N)$ of existing 2D-DTW approaches (see [LP92],[US98]).

3.4 Results

For the evaluation of our approach, we first describe the databases, metrics, comparisons and experiments performed.

Benchmarking Data

Two benchmarking databases of hand drawn symbols have been used, namely music symbols from musical scores, and architectural symbols from a sketching interface in a CAD framework. The first set is extracted from modern and old music scores, and it is used because of the high variability of the symbols, with important elastic deformations produced by the different writer styles. This database is fully described in the Appendix. The architectural database is used because it contains an important number of different classes with different appearance, while the inter-class variability is comparably lower.

Benchmarking Methods

Some benchmarking methods are chosen to compare our proposed features and our full DTW approach. The goal is to analyze the performance of our method but also the suitability of the set of features we propose.

Zernike moments [KH90] and a DTW cyclic method are used for comparing our DTW approach. Zernike moments are a classical shape method in the literature, and one of the MPEG-7 standards. They have been used in symbol recognition methods, because they are robust to deformations and invariant to scale and rotation. Zernike moments are defined over a set of complex polynomials which form a complete orthogonal set over the unit disk. In our experiments 7 moments are used.

We have also implemented a variation of our own method, named cyclic DTW. The idea is to see how the performance changes when using an algorithm with a lower computational cost. It consists of taking the center of mass of the symbol and for every orientation (from 0 to 180, with a step of 10 degrees) we only take into account the column that corresponds to the center of mass of the shape, and for this "centroid column", the features used in our approach are computed (the upper and lower profile, the sum of pixels per region). Thus, only one feature vector describes the symbol in every orientation. Then, a DTW cyclic approach (similar to a string matching cyclic) is used to match the matrices of the two symbols.

Concerning feature comparison, [RM03] and [MB01] features are compared against our features. In these experiments, our DTW approach has been applied using these features from the literature, which have been described in Section 3. Thus, we compare the proposed features against the ones defined by Rath and Marti to establish the suitability of our features.

Referring the method proposed in this paper, we use the upper and lower profiles, and the sum of pixels of 3, 4 or 5 regions. The features are extracted from every orientation, from 0 to 180 degrees, also with a step of 10 degrees.

Classification

For the classification of the symbols, one representative per class is usually chosen. Thus, every input symbol of the database is compared to these n representatives, and only n comparisons are computed for classifying every input symbol. Notice that with this approach, no training process is required, saving an important computational cost. The K-nearest neighbor (in our case, 1-NN) is used as the distance for the classification. The minimum distance will define the class where the input symbol belongs to.

3.4.1 Music Symbols Data Set

The data set of music clefs was obtained from a collection of modern and old musical scores (18th and19th centuries) of the Archive of the Seminar of Barcelona (an example can be seen in Figure 3.8(a)). This database contains a total of 2128 samples between the three different types of clefs from 24 different authors. The main difficulty of this database is the lack of a clear class separability because of the variation of the writer styles and the lack of a standard notation. The high variability

of clefs' appearance from different authors can be observed in the segmented clefs of Figure 3.8(b).

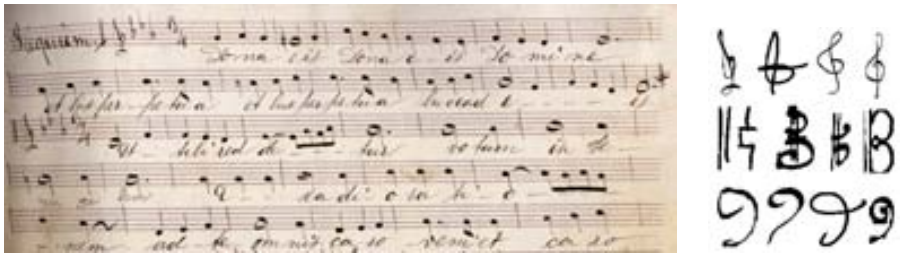


Figure 3.8: (a) Old musical score, (b) High variability of clefs appearance: first row shows treble clefs, second row shows alto clefs and the third one shows bass clefs.

Under this scenario, the selection of the representative for each class is not easy. The printed clefs that are shown in Figure 3.9(a),(b),(c) are not similar enough to the hand drawn ones. For this reason, we have chosen some hand drawn representative clefs: one treble clef (fig. 3.9(d)), one bass clef (fig. 3.9(e)), and two alto clefs (fig. 3.9(f)(g)) because of the high variability in alto clefs. The selected representatives correspond to the set median symbol.

(a) (b) (c) (d) (e) (f) (g)

Figure 3.9: Printed Clefs and Selected representative clefs: (a) Printed Treble clef. (b) Printed Bass clef. (c) Printed Alto clef. (d) Treble representative clef. (e) Bass representative clef. (f)(g) Two Alto representative clefs

Concerning the precision, recognition rate (recall) and fall-out (false positive rate) measures, they are computed using the following equations:

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)} \tag{3.12}$$

$$Recognition\ Rate = Recall = \frac{TruePositives}{Positives} \tag{3.13}$$

$$Fall - out = false\ positive\ rate = \frac{FalsePositives}{Negatives} \tag{3.14}$$

In table 3.1 the recognition rates of the classification of this data set are shown, where the DTW approach is compared to the Zernike moments, and DTW-cyclic, using the parameters defined above. One can see that with the method proposed we reach a recognition rate of 96.9%, significantly improving the Zernike moments (75.7%) and DTW-cyclic (65.5%) recognition rates.

Method	Zernike moments	DTW-cyclic	DTW-approach 5 zones
RR. Trebble Clef	87.7 %	27.1 %	96.2 %
RR. Bass Clef	63.8 %	91.4 %	96.5 %
RR. Alto Clef	75.7 %	78.0 %	97.1 %
Overall RR.	**75.7 %**	**65.5 %**	**96.6 %**
Overall Precision	**80.3 %**	**68.2 %**	**96.9 %**
Overall Fall-out	**11.9 %**	**19.6 %**	**1.8 %**

Table 3.1: Classification of clefs: Recognition Rate (RR.), Recall and Fall-out of these 3 music classes using 4 models.

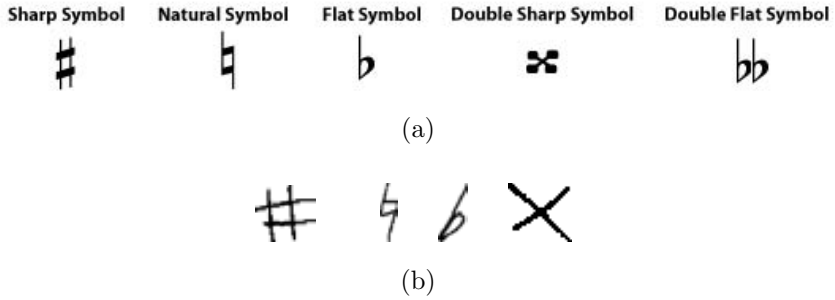
In table 3.2 we show the experimental results with some different features that can be used for describing the symbols. In this experiment, our DTW approach is always used, but making use of different features described in the literature, specifically those proposed by Rath and Marti. In table 3.2 we also show the recognition rates obtained using different numbers of regions (3, 4 and 5) in the feature extraction step of our approach. We can observe that Marti's features perform very well for the trebble and bass clefs (over 97% of recognition rate), but very poor with alto clefs (90%). Contrary, Rath's features achieve a good performance in alto clefs, but have some problems with trebble clefs. Concerning our features, we can see that the division of the image in 3 regions does not provide enough discriminatory power for the high variability in alto clefs (we reach a recognition rate of 94.3%), while the recognition rate increases when the number of regions is increased, reaching a 97.1% with 5 regions. In addition, it is shown that the features we have used achieve a better overall recognition rate and precision (96.6% and 96.9% respectively) in comparison to both of Marti (95% and 94.6%) and Rath's ones (96.1% and 96.5%), with a lower fall-out (1.8% in comparison to 2.6% of Marti and 2% of Rath's ones).

Features	Rath	Marti	DTW- 3z	DTW- 4z	DTW- 5z
Number of features per column	4	8	5	6	7
RR. Trebble Clef	95.8 %	97.3 %	96.7 %	96.3 %	96.2 %
RR. Bass Clef	96.1 %	97.6 %	96.5 %	96.3 %	96.5 %
RR. Alto Clef	96.5 %	90.1 %	94.3 %	96.1 %	97.1 %
Overall RR.	**96.1 %**	**95.0 %**	**95.8 %**	**96.2 %**	**96.6 %**
Overall Precission	**96.5 %**	**94.6 %**	**96.2 %**	**96.6 %**	**96.9 %**
Overall Fall-out	**2.0 %**	**2.6 %**	**2.2 %**	**2.0 %**	**1.8 %**

Table 3.2: Classification of clefs: Recognition Rates (RR.) of these 3 music classes using 4 models. Overall Recognition Rate (RR.), Precision and Fall-out of Rath's features, Marti's features and our DTW features, using 3, 4 and 5 regions (zones)

Clefs and Accidentals Data Set

An extension of these experiments has been performed including accidentals (sharps, naturals, flats and double sharps) in the musical symbol database. They are a total of 1970 accidentals drawn by 8 different authors. In Figure 3.10(a), one can see that some of them (such as sharps and naturals) can be easily misclassified due to their similarity. Contrary to double sharp, a double flat is just two flats drawn together, and for that reason double flats are not included in the accidentals' database.



(a)

(b)

Figure 3.10: Accidentals. (a) Printed accidentals appearing in music notation. (b) Selected representative accidentals: Sharp, Natural, Flat and Double Sharp models.

Similarly to the experiments previously showed, we have chosen one representative for each class (see Fig. 3.10(b)). The system will have now 8 models (4 clefs and 4 accidentals), and for every input symbol, 8 comparisons will be made. Results are shown in Table 3.12, where the DTW-based proposed descriptor reaches a 89.55% classification rate, outperforming the results obtained by the Zernike descriptor (43.97%).

3.4.2 Architectural Symbols Data Set

The architectural symbol data set is a benchmark database [SVL+04] comprising online and off-line instances from a set of 50 symbols drawn by a total of 21 users. Each user has drawn a total of 25 symbols and over 11 instances per symbol. Thus, the database (see examples in Fig.1.8) consists on 7465 individual instances, consisting of 50 symbols, each class with an average of 150 samples. It has been created with a Digital Pen & Paper framework [Log04]. To capture the data the following protocol has defined: The authors give to each user a set of 25 dot papers, which are paper containing the special pattern from Anoto. Each paper is divided into 24 different spaces where the user has to draw in. The first space is filled with the ideal model of the symbol to guide the users on their draw due to they are not experts on the field of Architectural design.

In this database the representative selected for each class (Fig. 3.11) corresponds to the printed symbol of the class, because both the printed and the hand drawn symbols are quite similar. The architectural symbol data set has been used to test the scalability of our method. In this experiment we test the performance under an increasing number of classes. We have started the classification using the first 5 classes. Iteratively, 5 classes have been added at each step and the classification

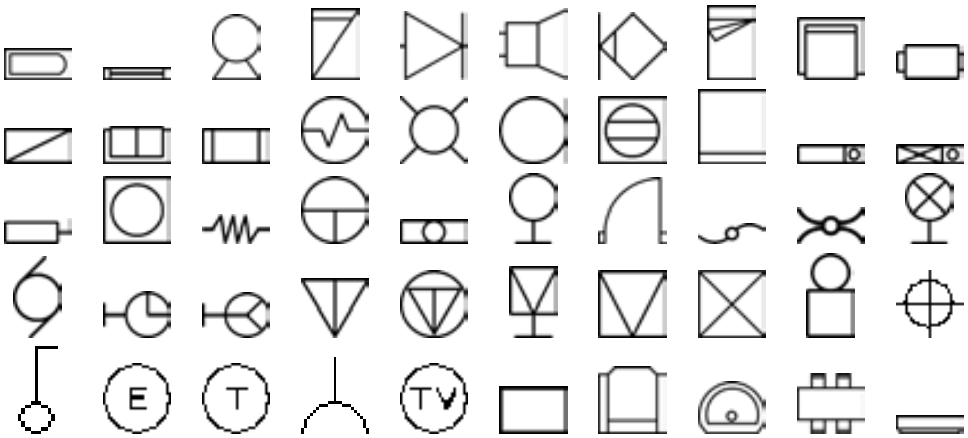


Figure 3.11: The fifty selected representatives for the architectural database.

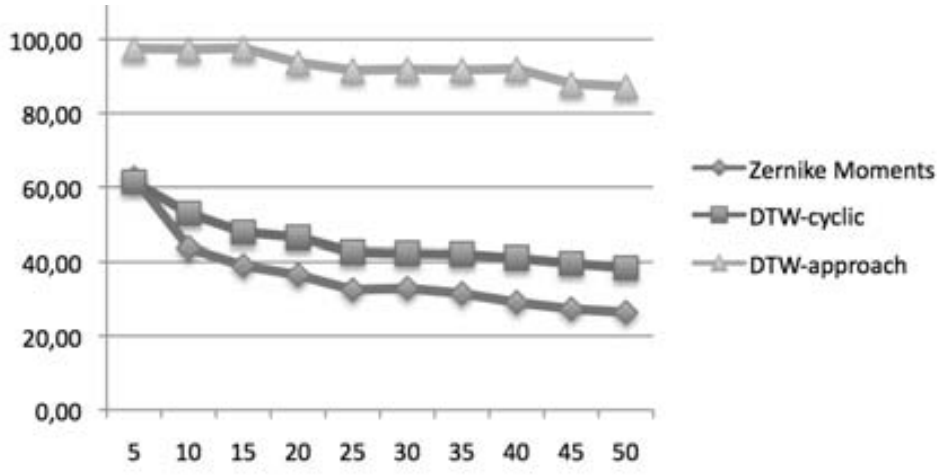


Figure 3.12: Classification of architectural hand drawn symbols to measure the scalability degree: Recognition rates using an increasing number of classes.

has been repeated. The more classes we introduce, the higher the confusion degree becomes among them. It is because of the elastic deformations inherent to hand drawn strokes, and the higher number of objects to distinguish. In Fig. 3.12 the recognition rates are presented, showing that our approach reaches significantly higher results than Zernike moments and the DTW-cyclic approach (87% in comparison to 26% and 38% respectively). The performance of the Zernike moments and the DTW-cyclic decrease dramatically when increasing the confusion in terms of the number of classes (Zernike momens decreases from 62% to 26% and DTW-cyclic decreases from 61% to 38% with 50 classes), whereas our method is quite robust to the increasing of the number of different classes participating (from 97% with 5 classes decreases to 87% with 50 classes).

3.5 Conclusions

We have presented a Dynamic Time Warping based method for the description and classification of hand drawn symbols. This approach is rotation and scale invariant, and robust to the deformations typical in hand drawn symbols. The method proposed computes a column sequence of feature vectors for each orientation of the two symbols and computes the DTW distance, taking also into account their perpendicular alignment. Our method has been tested with two hand drawn symbol databases (music and architectural) achieving high recognition rates. Comparison against some state-of-the-art descriptors shows the robustness and better performance of the proposed approach when classifying symbols with high variability in appearance, such as irregular deformations induced by hand drawn strokes, low inter-class and high intra-class variabilities.

The main drawback is the high computational cost: even though the method proposed is $O(w^2N^2)$, which is remarkably lower than other existing 2D-DTW approaches (such as $O(N^{4N})$ and $O(N^3 9^N)$), it is still not fast enough for performing symbol recognition in large databases or even real-time symbol recognition systems. In this sense, further work can be focused on developing DTW-variations for decreasing the time complexity of the algorithm.

In the next Chapter, a second symbol recognition method will be described. Both approaches will be used for the first writer identification approach proposed for writer identification in old music scores.

Chapter 4

The Blurred Shape Model descriptor for Symbol Recognition

Many symbol recognition problems require the use of robust descriptors in order to obtain rich information of the data. As it has been commented in the previous Chapter, the research of a good descriptor is still an open issue due to noise, deformations, occlusions and the high variability of symbols appearance. In this Chapter, we introduce another descriptor, namely Blurred Shape Model (BSM). It is a robust symbol descriptor which deals with most of these problematics. A symbol is described by a probability density function that encodes the probability of pixel densities of image regions. Afterwards, the Circular Blurred Shape Model (CBSM) is presented. It is an evolution of the BSM, which uses a correlogram structure for obtaining a rotation invariant descriptor. These descriptors have been evaluated on different hand-drawn and synthetic data sets, showing their robustness comparing it with the state-of-the-art descriptors.

4.1 Introduction

As it has been explained in Chapter 2, due to the kinds of problems in symbol recognition applications, some descriptors are better than others depending on the application field, and it is very common that symbol descriptors robust to some affine transformations and occlusions are not effective enough dealing with elastic deformations. For this reason, it is difficult to define an universal shape descriptor that suits in most fields. An ideal descriptor should guarantee intra-class compactness and inter-class separability, being tolerant to noise, degradation, occlusions, rotation, scaling, translation and non-uniform distortions appearing in hand-drawn symbols.

In the previous Chapter, a DTW-based symbol recognition method has been presented, which is specially defined for coping with deformations typically found in handwritten documents. In this Chapter we propose the Blurred Shape Model, a general descriptor that can deal with noise, degradation and occlusions. The descriptor encodes the spatial probability of appearance of the shape pixels and their context

information. As a previous step, the method aligns symbols' shape by means of the Hotelling transform and an area density adjustment. As a result, a robust technique in front of noise and elastic deformations is obtained.

Afterwards, the Circular Blurred Shape Model (CBSM) is presented. It is an evolution of the BSM descriptor, which not only copes with distortions and noise, but it is also rotation invariant. The CBSM codifies the spatial arrangement of object characteristics using a correlogram structure. By rotating the correlogram so that the major descriptor densities are aligned to the x-axis, the descriptor becomes rotation invariant. The presented methodologies are evaluated on synthetic and hand-drawn data sets. Different state-of-the-art descriptors are compared, showing the robustness and better performance of the proposed scheme when classifying large number of symbol classes with high variability of appearance.

The rest of the Chapter is organized as follows. First, the Blurred Shape Model (BSM) descriptor is described in Section 2. Secondly, the classification step is presented in Section 3. Experimental results are shown in Section 4. The Circular Blurred Shape Model (CBSM) is described in Section 5. Finally, concluding remarks are exposed in Section 6.

4.2 Blurred Shape Model

To describe a symbol that can suffer from irregular deformations, we propose to codify its *shape* by determining its *external appearance*. Here, we define the *external appearance pixels* as those which have high gradient magnitude. Taking into account those pixels, the Blurred Shape Model descriptor defines spatial regions where some parts of the symbol can be involved. For this task, the activated pixels (those set to one) from the input region to describe should belong to the shape of the symbol.

Given a shape image forming the shape $S = \{x_1, .., x_m\}$, we treat each point x_i, called from now SP, as a feature to compute the BSM descriptor of the symbol shape. The image region is divided in a grid of $n \times n$ equal-sized sub-regions (cells) r_i. Each cell receives votes from the SPs in it and also from the SPs in the neighboring sub-regions. Thus, each SP contributes to a density measure of its cell and its neighboring ones, and thus, the grid size identifies the blurring level allowed for the shape. This contribution is weighted according to the distance between the point and the center of coordinates c_i of the region r_i. The algorithm is summarized in table 2.

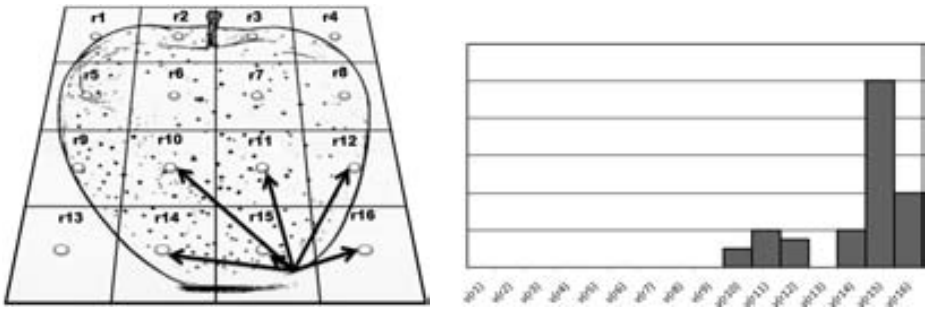
In Fig. 4.1, a shape description is shown for an apple data sample. Figure 4.1(a) shows the distances d_i of a SP to the nearest sub-regions centers. To give the same importance to each SP, all the distances to the neighbor centers are normalized. The output descriptor is a vector histogram v of length $n \times n$, where each position corresponds to the spatial distribution of SPs in the context of the sub-region and their neighbors ones. Fig. 4.1(b) shows the vector descriptor updating once the distances of the first point in Fig. 4.1(a) are computed. Observe that the position of the descriptor corresponding to the affected sub-region r_{15}, which centroid is nearest to the analyzed SP, obtains a higher value.

The resulting vector histogram, obtained by processing all SPs, is normalized in the range $[0, 1]$ to obtain the probability density function (pdf) of $n \times n$ bins.

Algorithm 2 Blurred Shape Model Description Algorithm.

Require: a binary image I, a number of regions r
Ensure: descriptor vector v

1: Obtain the *shape* S contained in I.
2: Divide I in $n \times n$ equal size sub-regions $R = \{r_1, ..., r_{n^2}\}$, with c_i the center of coordinates for each region r_i.
3: **Define** $N(r_i)$ as the neighbor regions of region r_i, defined as:
4: $N(r_i) = \{r_k | r_k \in R, ||c_k - c_i|| \leq 2|g|\}$, where g is the cell size.
5: **for** each point $\mathbf{x} \in S$ **do**
6: **for** each $r_i \in N(r_\mathbf{x})$ **do**
7: $d_i = d(\mathbf{x}, r_i) = ||\mathbf{x} - c_i||^2$
8: **end for**
9: Update the probability vector v as:
10: $v(r_i) = v(r_i) + \frac{1}{d_i D_i}, \quad D_i = \sum_{c_k \in N(r_i)} \frac{1}{||\mathbf{x} - c_k||^2}$
11: **end for**
12: Normalize the vector v as:
13: $v = \frac{v(i)}{\sum_{j=1}^{n^2} v(j)} \forall i \in [1, ..., n^2]$



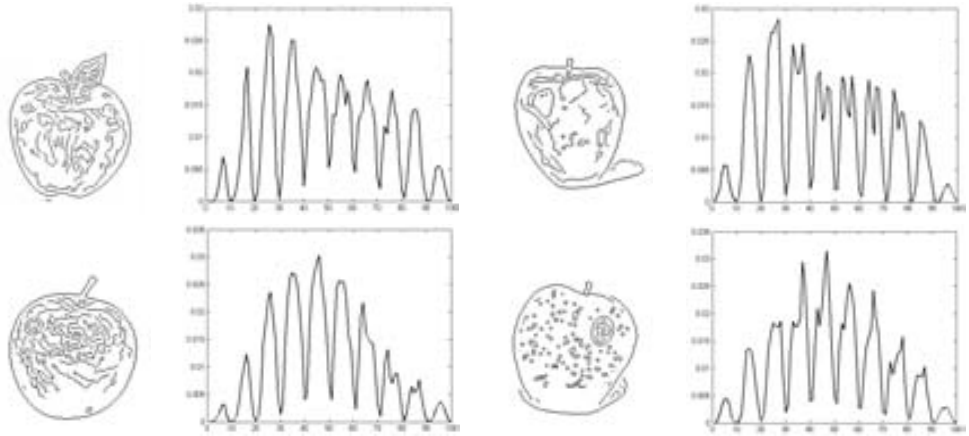
(a) (b)

Figure 4.1: Shape pixel distances estimation respect to neighbor centroids, and the vector actualization of the region 15th, where $\frac{1}{\sum distances} = 1$.

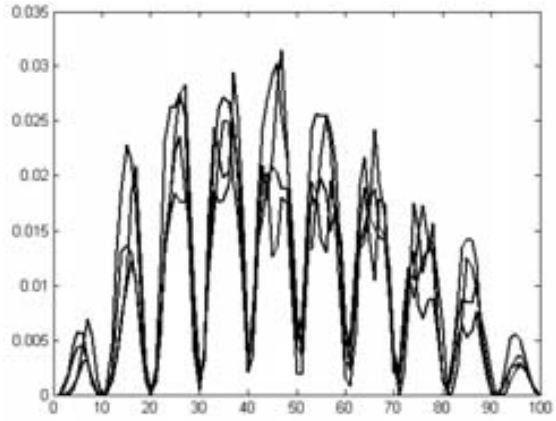
In this way, the output descriptor represents a distribution of probabilities of the symbol structure considering spatial distortions, where the distortion level allowed is determined by the grid size. The BSM descriptors for different grid sizes of an example of an architectural symbol are shown in Fig. 4.2. Concerning the computational complexity, for a region of $n \times n$ pixels, the k relevant considered SPs to obtain the BSM descriptor require a cost of $O(k)$ simple operations. In Fig. 4.3(a) four BSM descriptors of apple samples of length 10×10 are shown. Figure 4.3(b) shows the correlation of the four previous descriptors. Note that though it exists some variations on the shape of the symbols, the four descriptors remain closely correlated.

(a) (b) (c) (d) (e)

Figure 4.2: (a) Input image. (b) 48 regions blurred shape. (c) 32 regions blurred shape. (d) 16 regions blurred shape. (e) 8 regions blurred shape.



(a)



(b)

Figure 4.3: (a) Plots of BSM descriptors of length 10×10 for four apple samples. (b) Correlation of previous BSM descriptors.

4.3 Experimental Results

In order to validate the proposed methodology, first, we describe our performance evaluation protocol in terms of the used data, comparatives, and classification.

Data: To test the multi-class symbol recognition system, we used different scenarios: a hand-drawn music symbols data set, which main difficulty consists on the elastic deformations produced by the different writing styles; an architectural symbol database extracted from a sketching interface; and the GREC2005 database, a public printed symbols database with important distortions and noise.

Comparatives: The methods used in the comparative are: SIFT [Low04], Zoning, Zernike, ART and CSS curvature descriptors from the standard MPEG7 [KK99], [ZL04], [MM86]. The details of the descriptors used for the comparatives are the followings: The optimum grid size of the BSM descriptors is estimated applying cross-validation over the training set using a 10% of the samples to validate the different sizes of $n \in \{8, 12, 16, 20, 24, 28, 32\}$. For a fair comparison among descriptors, the Zoning descriptor is of the same length. The parameters for ART are radial order with value 2 and angular order with value 11. Concerning to Zernike, seven moments are used to estimate the descriptor, and a length of 200 with an initial sigma of one increasing per one is applied for the curvature space of the CSS descriptor. In order to deal with rotated symbols, before computing the BSM descriptor, the Hotelling transform based on principal components [Dun89] is applied to find the main axis of the object so the shape alignment can be performed.

Classification: To analyze the performance of the descriptors, we use 50 runs of Gentle Adaboost with decision stumps [FHT00], ten-fold cross-validation, and the one-versus-one ECOC design with the Euclidean distance decoding [PER08]. Although the focus of this Chapter is the proposal of a symbol descriptor, for the sake of performance evaluation, we will briefly introduce these two techniques. The Adaboost algorithm is applied to learn the descriptor features that best split classes, training the classifier from the descriptors. In this way, the Adaboost focuses on the discriminating regions by selecting the highest splitting features. In particular, we use the Gentle version of Adaboost since it has been shown to be dominant to the rest of variants when applied to real categorization problems [FHT00]. The Error Correcting Output Codes (ECOC) [DB95], [ETP+08] has been applied to deal with the multi-class categorization problem based on the embedding of binary classifiers. It is a meta-learning strategy that divides the multi-class problem in a set of binary problems, solves the individually, and aggregates their responses into a final decision. For a fair comparison, different base classifiers are used in the ECOC scheme: OSU implementation of Linear Support Vector Machines with the regularization parameter C set to 1 [OSU], OSU implementation of Support Vector Machines with Radial Basis Function kernel with the default values of the regularization parameter C and the gamma parameter set to 1 [OSU][1], and Linear Discriminant Analysis implementation of the PR Tools using the default values [PRT].

[1]The regularization parameter C and the gamma parameter are set to 1 for all the experiments. We selected this parameter after a preliminary set of experiments. We decided to keep the parameter fixed for the sake of simplicity and easiness of replication of the experiments, though we are aware that this parameter might not be optimal for all data sets.

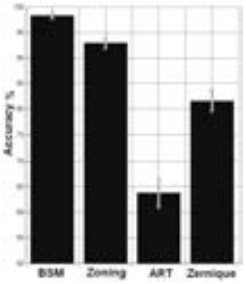
4.3.1 Music Symbols Data Set

The database of 2128 samples of clefs is obtained from a collection of modern and old musical scores of the Archive of the Seminar of Barcelona. Some examples of music clefs can be seen in Figure 3.8(b). This database, which has been described in the Appendix, has been also used for testing the DTW-based approach of the previous Chapter. It has been chosen for testing the robustness of the BSM descriptor in front of the high variability of symbols' appearance.

The accuracy and confidence ranges results for the old music clefs are shown and graphically represented in Fig.4.4(a) and Fig.4.4(b), respectively. ART and Zernike descriptors obtain the minor results, while the Zoning descriptor in the classification scheme technique offers good results. The BSM strategy is the most robust, obtaining an accuracy upon 98%.

Method	Accuracy
BSM	98.11 ± 0.59
Zoning	92.45 ± 1.06
ART	63.75 ± 2.76
Zernike	81.50 ± 1.97

(a)



(b)

Figure 4.4: (a) and (b) Clefs classification results.

Clefs and accidentals data set

An extension of these experiments has been performed including accidentals in the musical clef database. They are a total of 1970 accidentals drawn by 8 different authors, which has also been used for testing the DTW-based method. As a result, we obtain a database of 4098 samples from 7 different music symbols. A pair of segmented samples for each of the seven classes showing the high variability of clefs and accidentals appearance from different authors can be observed in Fig. 4.5.

Figure 4.5: Clefs and accidentals data set.

	FLDA	Linear SVM	RBF SVM	G. Adaboost	3NN
BSM	83.53(7.52)	80.51(7.31)	81.54(7.52)	**88.99(5.00)**	73.92(8.21)
Zoning	78.62(7.28)	79.45(6.30)	80.43(6.17)	**83.61(5.24)**	69.29(10.12)
SIFT	71.35(9.04)	**76.45(6.73)**	54.77(9.76)	74.95(9.77)	57.39(9.18)
CSS	68.76(11.02)	66.87(8.19)	69.87(9.18)	**71.33(8.44)**	61.28(8.92)
Zernike	69.09(6.01)	71.66(8.29)	59.21(9.00)	**72.05(7.76)**	54.12(9.10)

Table 4.1: Classification accuracy on the clefs and accidentals categories for the different descriptors and classifiers.

In order to classify this data set, we compare the BSM, Zoning, SIFT, CSS, and Zernique descriptors using the parameters defined above. Each feature set is learnt using the one-versus-one scheme with the previous commented base classifiers: FLDA, Linear SVM, RBF SVM, and Gentle Adaboost. Moreover, we include a comparative with a 3-Nearest Neighbor classifier to show the reliability of the present classification system. The performance and confidence interval obtained for each descriptor and classifier is shown in table 4.1. Looking at table 4.1, one can see that for each column corresponding to a different classifier, the descriptor that attains the best performance is the BSM. Moreover, looking the performances of each row corresponding to the results of each base classifier applied over each feature set, one can see that the base classifier that attains the best performance is the one-versus-one ECOC design with Gentle Adaboost as the base classifier, except in the case of the SIFT descriptor, which obtains its best performance with Linear SVM as the ECOC classifier. Finally, note that the results obtained by the 3NN classifier correspond to the lowest performance of each feature space.

4.3.2 Architectural Symbols Data Set

The database of architectural hand-drawn symbols has 2762 total samples organized in the 14 classes shown in Fig. 4.6. Each class consists of an average of 200 samples drawn by 13 different authors. This database is a subset of the database used for testing the DTW-based method, in which the 14 most representative symbols have been chosen. In this experiment, the architectural symbol database has been used to test the performance under an increasing number of classes, showing the scalability of our approach.

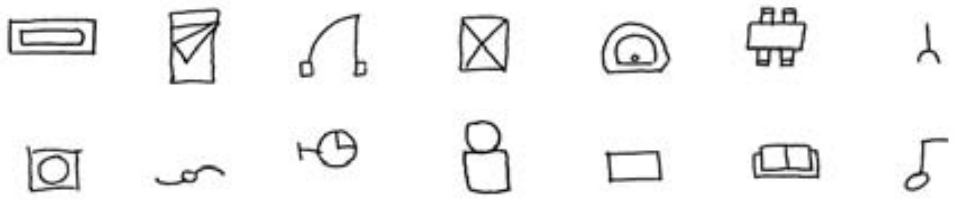


Figure 4.6: Architectural handwriting classes.

The results obtained from BSM are compared with the ART, Zoning, and Zernike moments. The compared descriptors are also introduced in the classification framework to quantify the robustness of each descriptor at the same conditions. Contrary to the DTW-based method experiments, we do not use the printed models for the classification. In this experiment, we use only the hand-drawn symbols for training and testing.

The classification starts using the first 3 classes. Iteratively, one class was added at each step and the classification is repeated. The higher number of classes, the higher confusion degree among them because of the elastic deformations inherent to hand drawn strokes, and the higher number of objects to distinguish. The results of accuracy recognition in terms of an increasing number of classes are shown in Fig. 4.7. The performance of the ART and Zernike descriptors decreases dramatically when we increase the confusion in terms of the number of classes, while Zoning obtains higher performance. Finally, the accuracy of the BSM outperforms the other descriptors results, and its confidence interval only intersects with Zoning in few cases. This behavior is quite important since the accuracy of the latter descriptors remains stable, and BSM can distinguish the 14 classes with an accuracy upon 90%.

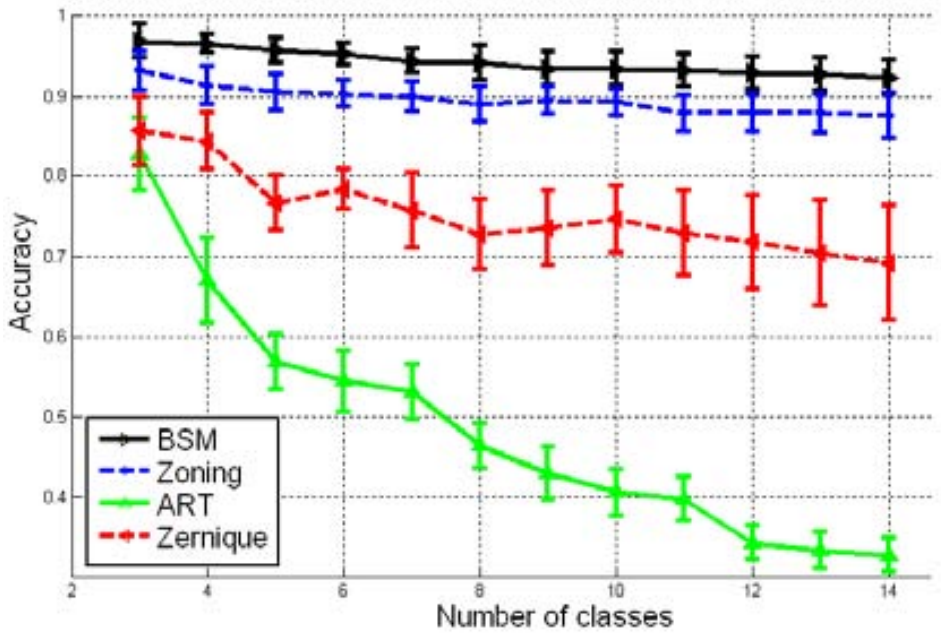


Figure 4.7: Descriptors classification accuracy increasing the number of architectural symbol classes (from 2 to 14 classes).

Referring the computational complexity, for a region of $n \times n$ pixels, the $k \leq n \times n$ skeleton points are considered to obtain the BSM with a cost of $O(k)$ simple operations, which is faster than the moment estimation of the ART and Zernike descriptors.

4.3.3 GREC-2005 Data Set

The GREC2005 database [2] [DV06] is not a hand drawn symbol database, but it has been chosen in order to evaluate the performance of our method on a standard, public and big database in front of distorted and noisy symbols. It must be said that our initial tests are applied on the first level of distortions (see Fig. 4.8). We generated 140 artificial images per model (thus, for each of the 25 classes) applying different distortions such as morphological operations, noise addition, and partial occlusions. In this way, the ECOC Adaboost is able to learn a high space of transformations for each class. The BSM descriptor uses a grid of 30×30 bins. In this sense, 900 features are extracted from every image, from which Adaboost selects a maximum of 50. For this experiment, we compare our results with the reported [ZLZ06] using the kernel density matching method (KDM). The results are shown in Table 4.2. One can see that the performances obtained with our methodology are very promising, outperforming for some levels of distortions the KDM results.

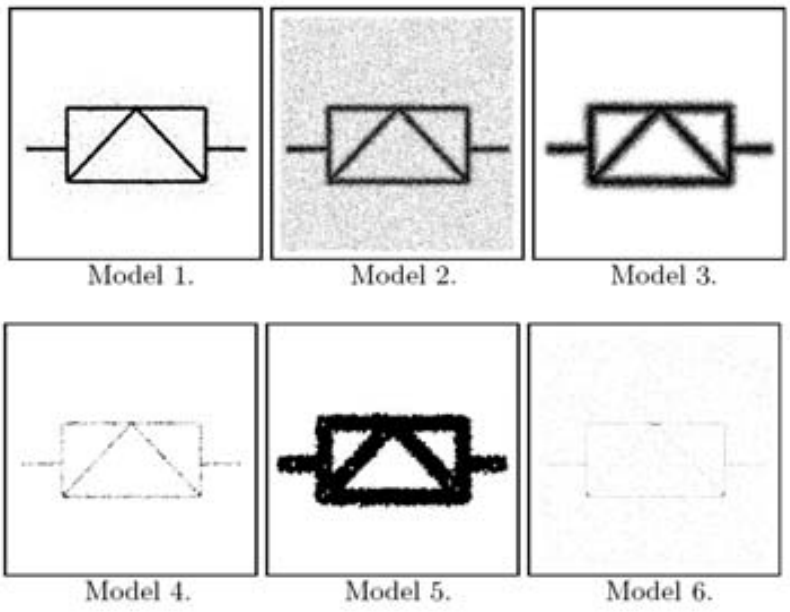


Figure 4.8: An example of the distortion levels used in the GREC2005 database.

4.3.4 Discussions

Concerning the suitability of the presented scheme to deal with multi-class symbol categorization problems, several benefits should be mentioned:

The method is rotation invariant because of the use of the Hottelling transform and the area density adjustment. The method is also scaling and (x, y) stretching

[2] http://symbcontestgrec05.loria.fr/formatgd.php

Method	Distort. Level 1	Distort. Level 2	Distort. Level 3	Distort. Level 4	Distort. Level 5	Distort. Level 6
KDM	100	100	100	96	88	76
BSM	100	100	100	100	96	92

Table 4.2: Descriptors classification accuracy increasing the distortion level of GREC2005 database using 25 models and 50 test images.

invariant because of the use of the $n \times n$ BSM grid. Moreover, the BSM descriptor is robust against symbols with rigid and elastic deformations since the size of the BSM grid defines the region of activity of the symbol shape points. The use of Adaboost as base classifier allows to learn difficult classes which may share several symbol features. Besides, the ECOC framework has the property of correcting possible classification errors produced by the binary classifiers, and allows the system to deal with multi-class categorization problems. When the classifiers are trained, only few features are selected, and when classifying a new test sample, only these features are computed. This makes the approach very fast and suitable for real-time categorization problems.

An important point of the BSM description is the selection of the grid size. The optimum size defines the optimum grid encoding the blurring degree based on a particular data set distortions. Because of this reason, a common way to look for the optimum grid size is applying cross-validation over the data for different descriptor parameters. In particular, we applied cross-validation using the 90% of the training subset samples, and the remaining 10% is used to validate the different possible sizes. The selected grid is the one which attains the highest performance on the validation subset, defining the optimum grid encoding the different distortions over each particular problem, and offering the required tradeoff between inter-class and intra-class variabilities in a problem-dependent way.

It is important to make clear that though Adaboost has been chosen as the base classifier in the presented system, depending on the problem we are working on, different alternatives of classifiers could be used instead, basing the selection of the base classifier on the type of distribution of the data and the behavior of each particular learning technique. Although at the previous experiments the comparative between Gentle Adaboost with ECOC and other state-of-the-art classifiers showed higher performance improvements of Adaboost, different results could be obtained over different data sets or with an exhaustive tuning of the parameters of the classifiers.

Moreover, it is important to mention different applications where the Multi-class BSM scheme could also be useful. Many description techniques are applied on problems where a previous region detection is required. As shown at the previous experiments, the BSM descriptor could be applied to this type of problems since it provides a fast and feasible way to robustly describe regions. In the same way, circular grids could also be defined to allow the BSM descriptor to be described on this type of applications, and also, for being rotation invariant without the need of the Hotelling transform. In this sense, the next section describes the proposed circular version of the BSM.

4.4 Circular Blurred Shape Model

In the previous section, the Blurred Shape Model (BSM) was presented. It is a descriptor that can deal with soft, rigid, and elastic deformations, but it is sensible to rotation. In this section, we present an evolution of the Blurred Shape Model descriptor, which not only copes with distortions and noise, but it is also rotation invariant. Feature extraction is performed capturing the spatial arrangement of significant object characteristics in a correlogram structure. Shape information from objects is shared among correlogram regions, where a prior blurring degree defines the level of distortion allowed to the symbol, making the descriptor tolerant to irregular deformations. The descriptor becomes rotation inavirant by rotating the correlogram so that the major descriptor densities are aligned to the x-axis. Moreover, the original BSM descriptor requires to align the object previously to its description, which considerably increases the computational cost in comparison to the proposed circular approach.

4.4.1 Circular Blurred-Shape Model

In this section, we present a circular formulation of the Blurred Shape Model descriptor. By defining a correlogram structure from the center of the object region, spatial arrangement of object parts is shared among regions defined by circles and sections. The method aims to achieve a rotation invariant description rotating the correlogram by the predominant region densities, which implies the full redefinition of the BSM descriptor. We divide the description of the algorithm into three main steps: the definition of the correlogram parameters, the descriptor computation, and the rotation invariant procedure.

Correlogram definition: Given a number of concentric circles C, a radius length R, a number of sections S, and an image region I, a centered correlogram $B = \{b_{\{1,1\}}, .., b_{\{C,S\}}\}$ is defined as a radial distribution of sub-regions of the image, as shown in Figure 4.9(a) and (b). Each region b defines a centroid coordinates b^* (see Fig. 4.9(c)). Then, the regions around b are defined as the neighbors of b. Note that depending of the spatial location of the analyzed region, different number of neighbors can be defined (see Fig. 4.9(d)). Different correlogram structures are shown in Figure 4.10 for different values of C and S.

Descriptor computation: In order to compute the CBSM descriptor, first, a pre-processing of the input region I to obtain the shape features is required. For several symbols, relevant shape information can be obtained by means of a contour map (although based on the object properties we can define a different pre-processing step). In this section, we use the Canny edge detector procedure.

Given the object contour map, each point from the image belonging to a contour is taken into account in the description process (see Fig. 4.9(e)). First of all, the distances from the contour point $\mathbf{x}$ to the centroids of its corresponding region and neighboring regions are computed. The inverse of these distances are computed and normalized by the sum of total distances. These values are then added to the corresponding positions of the descriptor vector ν (see Fig. 4.9(f)). This makes the description tolerant to irregular deformations. Concerning the computational com-

plexity, note that for a correlogram of $C \times S$ sectors and k contour points considered for obtaining the CBSM descriptor, only $O(k)$ simple operations are required. The description procedure is detailed in Algorithm 3.

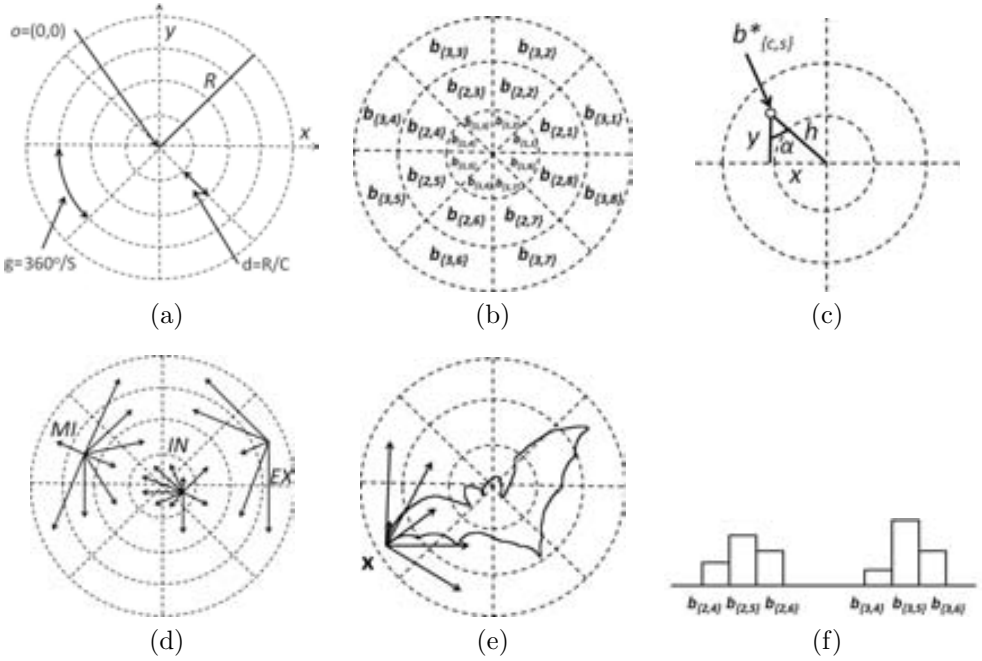


Figure 4.9: (a) CBSM correlogram parameters, (b) regions distribution, (c) region centroid definition, (d) region neighbors, (e) object point analysis, and (f) descriptor vector update after the analysis of point x.

At this point we have a description ν for an input region I, where the length of ν, defined by parameters C, S, and R, defines the degree of spatial information taken into account in the description process. In Figure 4.11, a bat instance from the public MPEG7 data set [MPE] is described with different $C \times S$ correlogram sizes. Note that when we increase the number of regions, the description becomes more local. Thus, optimal parameters of C and S should be obtained for each particular problem (e.g. via cross-validation, splitting the training data into two subsets, one to train and the remaining one to validate the method parameters).

Rotation invariant descriptor: For obtaining a rotation invariant descriptor, a second step is included in the description process. We look for the main diagonal G_i of correlogram B which maximizes the sum of the descriptor values at affected sectors. This diagonal is then taken as reference to rotate the descriptor. The orientation of the rotation process, so that G_i is aligned to the x-axis, is that one corresponding to the highest density of the descriptor at both sides of G_i. This procedure is detailed in Algorithm 4. A visual result of the rotation invariant process can be observed in Fig. 4.11, in which two bats with different orientations are rotated and aligned.

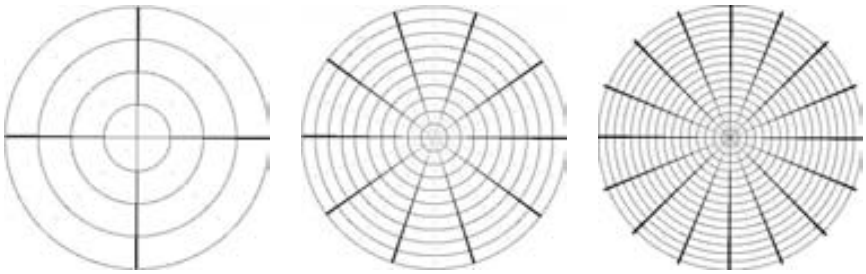


Figure 4.10: Correlogram structures obtained for different $C \times S$ sizes: (a) 4×4, (b) 10×10, and (c) 16×16.

The CBSM correlogram is defined by means of a number of sectors S and number of concentric circles C in a linear correlogram design. It implies that the area of the external sectors is higher than the area corresponding to inner sectors. Since we define the same importance to all analyzed shape points, it seems intuitive to define sectors with the same area. However, in this paper we define a linear concentric circles definition which implies more local description on the center of the description meanwhile the distortion degree allowed at the external sectors is increased. We use this approximation based on the fact that the external appearance of symbols is usually higher compared to the inner variabilities (i.e. the external strokes in hand-drawn symbols). We also apply cross-validation in order to estimate the optimum C and S parameters based on each particular data set. On the other hand, if we want to define a correlogram structure where all sectors have the same area, we simply need to change the distance among correlogram sectors to satisfy the new constraints.

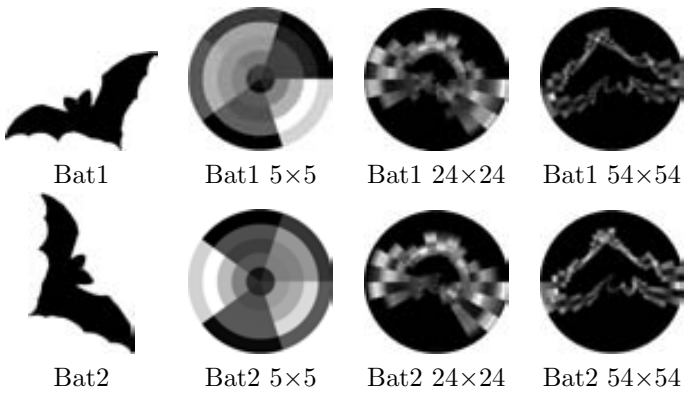


Figure 4.11: Examples of image descriptors at different sizes for two object instances. The two descriptors are correctly rotated and aligned.

Algorithm 3 Circular Blurred Shape Model Description Algorithm.

Require: a binary image I, a length of radius R, a number of concentric circles C, and a number of sections S

Ensure: descriptor vector ν

Define $d = R/C$ and $g = 360/S$ as the distance between consecutive concentric circles and the degrees between consecutive sectors, respectively (Figure 4.9(a)).

Define $B = \{b_{\{1,1\}}, .., b_{\{C,S\}}\}$ as the set of bins for the circular description of I, where $b_{c,s}$ is the bin of B between distances $[(c-1)d, c \cdot d)$ to the origin of coordinates o, and between interval angles $[(s-1)g, s \cdot g)$ to the origin of coordinates o and x-axis (Figure 4.9(b)).

Define $b^*_{\{c,s\}} = (\sin \alpha\ d, \cos \alpha\ d)$ the centroid coordinates of bin $b_{\{c,s\}}$, and $B^* = \{b^*_{\{1,1\}}, .., b^*_{\{C,S\}}\}$ the set of centroids in B (Figure 4.9(e)).

Define $X_{b_{\{c,s\}}} = \{b_1, .., b_{c \cdot s}\}$ as the sorted set of the elements in B^* so that $d(b^*_{\{c,s\}}, b^*_i) \leq d(b^*_{\{c,s\}}, b^*_j)$, $i < j$.

Define $N(b_{\{c,s\}})$ as the neighbor regions of $b_{\{c,s\}}$, defined by the initial elements of $X_{b_{\{c,s\}}}$:

$$N(b_{\{c,s\}}) \quad = \quad \begin{cases} X', |X'| = S+3 & \text{if } b_{\{c,s\}} \in IN \\ X', |X'| = 9 & \text{if } b_{\{c,s\}} \in MI \\ X', |X'| = 6 & \text{if } b_{\{c,s\}} \in EX \end{cases}$$

being IN, MI, and EX, the inner, middle, and extern regions of B, respectively (Figure 4.9(c)).

Initialize $\nu_i = 0$, $i \in [1, .., C \cdot S]$, where the order of indexes in ν are:

$\nu = \{b_{\{1,1\}}, .., b_{\{1,S\}}, b_{\{2,1\}}, ..b_{\{2,S\}}, .., b_{\{C,1\}}, ..b_{\{C,S\}}\}$

for each point $\mathbf{x} \in I$, $I(\mathbf{x}) = 1$ (Figure 4.9(d)) **do**

  $D = 0$

  for each $b_i \in N(b_{\mathbf{x}})$ **do**

    $d_i = d(\mathbf{x}, b_i) = ||\mathbf{x} - b^*_i||^2$

    $D = D + \frac{1}{d_i}$

  end for

  Update the probabilities vector ν positions as follows (Figure 4.9(f)):

  $\nu(b_i) = \nu(b_i) + \frac{1}{d_i D}$, $\forall i \in [1, .., C \cdot S]$

end for

Normalize the vector ν as follows:

$d' = \sum_{i=1}^{C \cdot S} \nu_i$, $\nu_i = \frac{\nu_i}{d'}, \forall i \in [1, .., C \cdot S]$

Algorithm 4 Rotation invariant ν description.

Require: ν, S, C

Ensure: Rotation invariant descriptor vector ν^k

Define $G = \{G_1, .., G_{S/2}\}$ the $S/2$ diagonals of B, where $G_i = \{\nu(b_{\{1,i\}}), .., \nu(b_{\{C,i\}}), .., \nu(b_{\{1,i+S/2\}}), .., \nu(b_{\{C,i+S/2\}})\}$

Select G_i so that $\sum_{j=1}^{2C} G_i(j) \geq \sum_{j=1}^{2C} G_k(j)$, $\forall k \in [1, .., S/2]$

Define L_G and R_G as the left and right areas of the selected G_i as follows:

$L_G = \sum_{j,k} \nu(b_{\{j,k\}}, j \in [1, .., C], k \in [i+1, .., i+S/2-1]$

$R_G = \sum_{j,k} \nu(b_{\{j,k\}}, j \in [1, .., C], k \in [i+S/2+1, .., i+S-1]$

if $L_G > R_G$ **then**

  B is rotated $k = i + S/2 - 1$ positions to the left:

  $\nu^k = \{\nu(b_{\{1,k+1\}}), .., \nu(b_{\{1,S\}}), \nu(b_{\{1,1\}}), .., \nu(b_{\{1,k\}}), ..,$

  $, .., \nu(b_{\{C,k+1\}}), .., \nu(b_{\{C,S\}}), \nu(b_{\{C,1\}}), .., \nu(b_{\{C,k\}})\}$

else

  B is rotated $k = i - 1$ positions to the right:

  $\nu^k = \{\nu(b_{\{1,S\}}), .., \nu(b_{\{1,S-k+1\}}), \nu(b_{\{1,1\}}), .., \nu(b_{\{1,S-k\}}), ..,$

  $, .., \nu(b_{\{C,S\}}), .., \nu(b_{\{C,S-k+1\}}), \nu(b_{\{C,1\}}), .., \nu(b_{\{C,S-k\}})\}$

end if

4.4.2 Experimental Evaluation

In order to present the multi-class categorization results, we discuss the data, methods, and validation of the experiments:

• *Data*: For comparing our CBSM multi-class methodology, we have chosen the public 70-class MPEG7[3] binary repository data set [MPE], which contains a high number of classes with different appearance of symbols from a same class, including rotation.

• *Methods*: The descriptors considered in the comparative are SIFT [Low04], BSM, Zoning, Zernike, and CSS descriptors from the standard MPEG7 [KK99], [ZL04], [MM86]. The details of the descriptors used for the comparatives are the following: the optimum correlogram size of the CBSM descriptor is estimated applying cross-validation over the training set using a 10% of the samples to validate the different sizes of $S = \{8, 12, 16, 20, 24, 28, 32\}$ and $C\{8, 12, 16, 20, 24, 28, 32\}$. For a fair comparison among descriptors, the Zoning and BSM descriptors are set to the same number of regions as the CBSM descriptor. Rotation invariance for the BSM descriptor is achieved by means of principal components alignment (using the Hotelling transform) before descriptor computation. Concerning the Zernike technique, 7 moments are used. The length of the curve for the CSS descriptor is normalized to 200, where the sigma parameter takes an initial value of 1 and increases by 1 unit at each step (experimentally tested). Gentle Adaboost with 50 decision stumps [FHT00] is used to train the binary problems of the one-versus-one ECOC design [ETP+08] to solve the multi-class categorization problems. We also consider a Support Vector Machine with a Radial Basis Function base classifier for the ECOC design with $C = 1$ and $\gamma = 1$[4] and a 3-Nearest Neighbor classifier in the comparative.

• *Validation*: The classification score is computed by means of stratified ten-fold cross-validation, testing for the 95% of the confidence interval with a two-tailed t-test.

Next, we describe the experiments performed, comparing our descriptor with state-of-the-art descriptors over two multi-class categorization problems (with binary and grey-level symbols).

MPEG7 data set

The MPEG7 data set [MPE] has been chosen since it provides a high intra-class variability in terms of scale, rotation, rigid and elastic deformations, as well as a low inter-class variability. It contains 70 different classes, thus it can be used to test the performance of the methods in front of a high number of classes. A pair of samples for some categories of the data set are shown in Fig. 4.12. Each of the classes contains 20 instances, which represents a total of 1400 symbol samples for the 70 classes.

In order to classify the data set, we compare the BSM, Zoning, SIFT, CSS, and Zernike descriptors with the previous defined parameters. Each feature set is learnt using the one-versus-one scheme with Gentle Adaboost. We include a comparative

[3]MPEG7 Repository Database: http://www.cis.temple.edu/ latecki/research.html

[4]As in the BSM experiments, the regularization parameter C and the γ parameter are set to one for the experiments. We selected this parameter after a preliminary set of evaluations. We decided to keep the parameter fixed for the sake of simplicity and easiness of replication of the experiments, though we are aware that this parameter might not be optimal for the analyzed data sets.

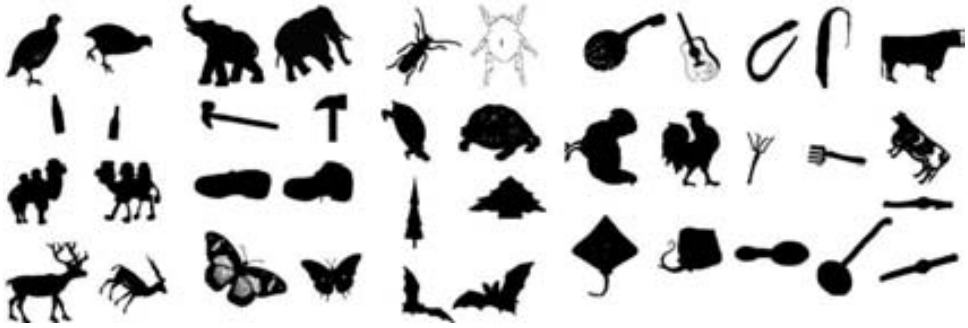


Figure 4.12: MPEG7 data set. Two examples of several classes are shown.

Descriptor	$3NN$	ECOC & Adaboost
CBSM	**71.84(6.73)**	**80.36(7.01)**
BSM	65.79(8.03)	77.93(7.25)
Zernike	43.64(7.66)	51.29(5.48)
Zoning	58.64(10.97)	65.50(6.64)
CSS	37.01(10.76)	44.54(7.11)
SIFT	29.14(5.68)	32.57(4.04)

Table 4.3: Classification accuracy on the 70 MPEG7 symbol categories for the different descriptors using 3-Nearest Neighbor and the one-versus-one ECOC scheme with Gentle Adaboost.

with a 3-Nearest Neighbor classifier to show reliability of the present classification system. The performance and confidence interval obtained by the six descriptors and the two classifiers is shown in table 4.3.

Having a look at the results obtained, one can see that for each descriptor, the Adaboost with ECOC approach always obtains higher performance than classifying with a nearest neighbor classifier in this data set. In addition, and for a same classifier, the difference among descriptor performances are more significant. It is produced because of the high number of classes and the high variability of appearance of the symbols shape. The best performance is obtained by our CBSM descriptor (about 80%), followed by the BSM descriptor (about 78%). The Zoning and Zernique moments obtain a recognition rate of 65% and 51% respectively, being remarkably lower than the CBSM and BSM descriptors. Finally, the CSS and SIFT descriptors obtain the worst recognition rates (under the 45%). The performance of these two last descriptors is expected since they focus on the points of curvature from the symbols shape and the degrees of orientation from the image derivatives, which significantly changes in this data set for the samples of a same class.

4.5 Conclusions

In this Chapter, we have presented the Blurred Shape Model descriptor and the Circular Blurred Shape Model, an evolution of the first one. The Blurred Shape Model is a simple descriptor that in a fast way defines a probability density function of the shape of a symbol. The shape is parameterized with a set of probabilities that encode the spatial variability of the symbol, being robust to several symbol distortions. Secondly, the Circular Blurred Shape Model descriptor has been presented. The descriptor encodes the spatial arrangement of symbol characteristics using a correlogram structure. A prior blurring degree defines the level of degradation allowed to the symbol. Moreover, the descriptor correlogram is rotated guided by the major density, becoming rotation invariant.

The approaches have been evaluated on different hand-drawn and synthetic data sets. Different state-of-the-art descriptors are compared, showing the robustness and better performance of the proposed scheme when classifying symbols with high variability of appearance, such as occlusions, rigid or elastic deformations, gaps or noise. In particular, the performance improvements are more significant when the described symbols suffer from irregular deformations. Concerning the time complexity, the BSM and CBSM descriptors are fast to compute, thus, they are suitable for real-time applications and for symbol detection problems.

Concerning the symbol recognition method described in the previous Chapter, a comparison with the BSM and CBSM should be performed. It must be said that the DTW-based method uses the DTW algorithm for computing the distance between two symbols, and k-NN is used for the classification, avoiding the training step. Contrary, the BSM and CBSM approaches use the Adaboost and ECOC framework for obtaining high recognition rates. As an example, for the clefs and accidentals music database, the BSM obtains a 89% of recognition rate using the Adaboost & ECOC, whereas the rate decreases to 74% using k-Nearest Neighbour. Notice that this value is remarkably lower than the DTW-based method, which obtains a 89.5% of recognition rate. In this sense, the DTW-based method is more suitable for the recognition of hand-drawn symbols with a high variability because of the different writer styles than the BSM approach.

The main advantage of the BSM and CBSM descriptors is that they are general descriptors useful in different scenarios. Whereas the DTW-based approach is focused on the problematic of high variability appearance (obtaining very good results), the BSM can reach good performance in front of the most problematics of symbol recognition. As an example, the DTW-based method is sensible to noise and gaps, because the upper and lower profiles will be very affected. Contrary, the BSM/CBSM can effectively deal with this kind of problems, reaching very high recognition rates (about 92% of recognition rate on the distortion level 6 of the GREC 2005 database). As a summary, depending on the problematic of the symbol recognition problem, the user will choose the DTW-based method or the BSM/CBSM descriptors.

Chapter 5

A Symbol-Dependent Writer Identification Approach Based on Symbol Recognition Methods

Writer identification consists in determining the writer of a piece of handwriting from a set of writers. Even though an important amount of compositions contains handwritten text in the music scores, the aim of this thesis is to use only music notation to determine the author. In this chapter we introduce a *symbol-dependent* approach for identifying the writer of a music score, which is based on the symbol recognition methods described in Chapter 3 and 4. The main idea is to use the BSM descriptor and the DTW-based method for detecting, recognizing and describing the music clefs. The proposed approach has been evaluated in a database of old music scores, achieving very good writer identification rates.

5.1 Introduction

As we stated in the Introduction, there are two major approaches for writer identification, namely text-dependent and text-independent. When dealing with graphical information, we referred to the above concepts as *symbol-dependent* and *symbol-independent*. In this Chapter, a *symbol-dependent* writer identification method is proposed, which combines the two symbol recognition methods proposed in the previous Chapters (the DTW-based method and the Blurred Shape Model) for detecting and extracting features of some specific symbols. The idea is to detect these symbols, and then perform writer identification based on the information extracted of the symbols' shape.

In the Introduction, we have described the discriminant properties of the handwriting style in music notation. After analyzing the different music elements and their characteristics, the following points are concluded. First, the properties about the staff lines have been discarded for our writer identification approach because there

is a big amount of printed staff lines. Second, the discrimination of bar lines is low, and the probability of existing accidentals, rests and ending signatures in every music sheet is not high, thus they are not taken into account. Third, the differences between the writing style of music notes are reduced when increasing the number of writers, and consequently, the discrimination power becomes low. In any case, they can be used in combination with other characteristic properties. Concerning lyrics, it must be said that although the identification of the writer using text in this work has not been considered, not all the music scores contain text (e.g. music scores for instruments), and in addition, in some cases, the writer of the lyrics and the writer of the music notation is not the same. There are also some limitations for using information extracted from dynamics, tempo markings and time signature. Firstly, there is a high number of different dynamic, tempo markings and time signature, and secondly, the probability to find the same indication in different music sheets is very low. Referring music, clefs, they can be seen as a characteristic individual signature of a writer, having a high discrimination power. An important advantage is that there are only three different clefs (alto, bass or trebble clef) to consider. In addition, clefs are usually appearing in each music sheet, allowing the comparison between music scores.

For these reasons, we have focused on the extraction of properties of music clefs. A similar idea has been used in the text-dependent approach for writer identification in Hebrew documents [BYBKD07] reviewed in the state of the art. This method is based on the detection and extraction of features from three pre-defined Hebrew characters (Aleph, Lamed, Ain). The rest of the characters are not taken into account for the classification. Having a look at the Hebrew characters (see Fig.2.2), one can see that they could be treated as symbols.

Our proposed *symbol-dependent* method detects and recognizes the music clefs and then, it performs writer identification based on the shape descriptors computed from each clef. Two main tasks are addressed here. The first one is related to the clef detection, whose aim is the localization and segmentation of the music clef in the image, discarding the other symbols. The second one is related to the clef description, whose aim is the characterization and description of the clef in order to classify the symbol to its corresponding true class given a set of possible classes. For this second task, we require a robust descriptor able to cope with hand-drawn distortions and also with the inaccuracy on the clef segmentation.

The remainder of the Chapter is structured as follows. In the next section, the preprocessing steps are presented, in which the music score is binarized and the staff lines are removed. In Section 3 the clef detection technique is fully described, which combines the BSM and the DTW-based methods. In Section 4 the description and classification of music clefs is presented. Experimental results are shown in Section 5. Finally, Section 6 concludes the Chapter.

5.2 Preprocessing

The preprocessing phase consists in binarizing the image, deskewing it and removing the staff lines. These process is fully described in Chapter 8, where the complete application scenario is described, and will be briefly commented next.

In the first step the gray-level scanned image (at a resolution of 300 dpi) is binarized to separate foreground from background. The second step consists in deskewing the image, so that staff lines would be horizontal and their recognition will be easier. The Hough Transform method has been used for detecting the staff lines, and for obtaining the rotation angle (in case the deskewing is necessary).

The third step consists in removing the staff lines. As it has been commented in Chapter 1, staff lines play a central role in music notation because they define the vertical coordinate system for pitches and give a size normalization useful for symbol recognition (size of musical symbols is linearly related to the staff space). Unfortunately, staff causes distortions in musical symbols (connecting objects that should be isolated), making difficult the recognition process. For that reason, staff removal is performed in order to isolate musical symbols. In case of old handwritten music scores, the staff removal process must cope with paper degradation, the warping effect, distortions and gaps. The method proposed consists in obtaining a coarse approximation of the staff lines applying projections and median filters with a horizontal mask, Then, the staff is reconstructed joining these segments depending on the orientation, distance and area of each segment. Finally, a contour tracking process is used for following and removing every staff line, taking into account the coarse approximation when gaps are appearing. In this stage, the staff lenght (which corresponds to the size of the five staff lines), the staff line width and the staff line distance (which corresponds to the distance between two consecutive staff lines) are computed.

5.3 Clef Detection and Segmentation

The method proposed for writer identification is composed of two tasks, namely, clef detection and clef description. The first step consists in detecting the music clefs in the music score. We have formalized it as a symbol detection problem [TTD06]. The aim of symbol detection is the localization of some important information instead of analyzing the whole content of the document, because of the following reasons. First, the recognition of the whole document can be a very complex task (e.g. the analysis of historical documents); and secondly, a fast symbol detection technique is required for localizing symbols in large data sets. One can note the *chicken & egg* problem as the *segmentation-recognition* paradox, because we can not decide between segmenting for recognizing and recognizing for segmenting, being the ideal solution to perform both tasks at the same time. Symbol detection is related to indexing and retrieval, and it has been a very emerging topic of interest, applied to graphic documents such as technical drawings [SM99] or maps [SS96]. The detection techniques can rely on different pattern recognition methods, such as the geometric features described in [FJ03], the region-based approach using connected components [RnL08], signatures with look-up tables [RnLS09], or the structural symbol representation [ZT06].

A symbol detection method requires a good localization strategy and a robust symbol descriptor. Concerning the localization step, the aim is to localize the target symbol while discarding the most part of the image. In addition, it is important to avoid the analysis of the whole image with a sliding window for saving time. Referring

the detection step, the descriptor should cope with deformation, distortions, noise and segmentation. It should be said that for obtaining characteristics of the music clefs, it is not necessary to detect all the clefs of the image. Contrary to Optical Music Recognition, badly segmented or incomplete music clefs could be left out, in order to avoid the introduction of noise to the classification step.

For our clef detection method, we use a combination of the BSM descriptor and the DTW-based symbol recognition method (described in the previous Chapters), because they have shown to be robust descriptors, able to cope with the irregular deformations typically found in hand-drawn symbols.

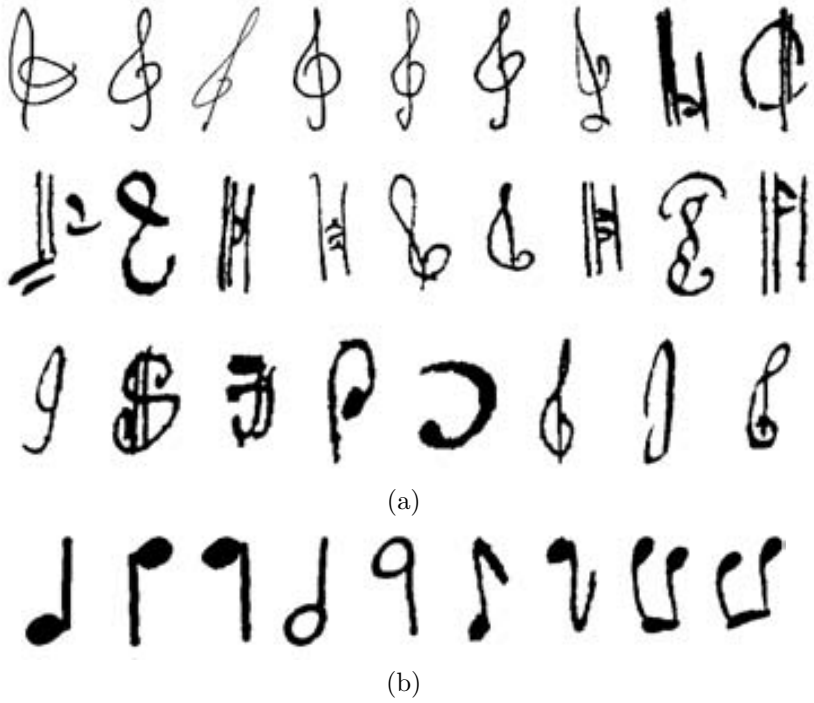
In order to design a symbol detection methodology, we need to define two stages. A first stage should learn to distinguish among the target symbol and the background (e.g. learning a binary classifier). A second stage should perform a search over the whole image using the trained classifier in order to locate those regions containing the target symbol.

5.3.1 Training Process

For the first step, we propose to learn a hierarchical cascade of 2 classifiers with a set of positive and negative clef instances, manually extracted from a set of music scores (an example of the positive clefs and negative examples used can be seen in Fig.5.1). Initially, the set of positive samples consists in clefs extracted from the music scores, whereas the negative examples are basically, examples of music notes. In the training stage, the suitable parameters of the BSM and the DTW-based descriptors are found, and the set of negative examples can be modified. First, different grid sizes and the rejection threshold for the BSM descriptor are tested until a minimum accuracy is achieved. Then, the set of negative examples is modified, adding the images of the false detections found. Secondly, different number of features (the number of regions) and the rejection threshold for the DTW-based symbol recognition method are tested. Finally, the set of false detections can be also increased by adding the images of the false detections found. This strategy is detailed in Algorithm 5.

5.3.2 Detection Process

Once both classifiers are trained, the different elements must be segmented from the input image. For this purpose, the graph contraction process is used, which consists in applying a morphological dilate using disks of different sizes as the structuring element. Then, the connected components whose size and area are not under certain restrictions (no clef is smaller than the half of the staff length and bigger than twice the staff length) are removed. This step is used for discarding the too small or too big symbols, which consequently, are not music clefs. Afterwards, the BSM descriptor is computed for each remaining connected component, and compared with the BSM descriptors of the set of positive and negative examples. The comparison is performed using the Euclidean distance and the k-NN classifier. If the BSM-based classifier accepts the candidate connected component as a clef, then, the DTW-based features are computed for this region, and compared with the DTW-based features of the set of positive and negative examples. If the DTW-based classifier also accepts the



(a)

(b)

Figure 5.1: Example of sets of images used in the training step. (a) Positive images of clefs. (b) Negative images of clefs (music notes).

Algorithm 5 Symbol Detection Training algorithm for the cascade of two classifiers.

Require: A set of positive examples P and a set of negative examples N, a maximum false alarm rate f, a minimum accuracy a.

$F_i \leftarrow 1$, $n_i \leftarrow 0$
while $F_i > f$ **do**
  $n_i \leftarrow n_i + 1$
  Use P and N to train a classifier using the BSM descriptor with n_i as the grid size
  $F_i \leftarrow$ Evaluate current classifier on validation set
  Decrease threshold for the ith classifier until the current cascaded classifier satisfies a detection rate of a (this also affects F_i)
end while
$N \leftarrow 0$
Evaluate the BSM-based detector on the set of non-symbol images and put any false detections into the set N.
$F_i \leftarrow 1$, $n_i \leftarrow 0$
while $F_i > f$ **do**
  $n_i \leftarrow n_i + 1$
  Use P and N to train a classifier using the DTW-based method with n_i regions
  $F_i \leftarrow$ Evaluate current classifier on validation set
  Decrease threshold for the ith classifier until the current cascaded classifier satisfies a detection rate of a (this also affects F_i)
end while
Evaluate the DTW-based detector on the set of non-symbol images and put any false detections into the set N.

Ensure: A cascade h of the BSM and DTW-based classifiers for symbol detection.

candidate connected component as a clef, then the candidate region is accepted as a music clef. The method is described in Algorithm 6.

Algorithm 6 Symbol detection using a cascade of two classifiers.

Require: An image I, a cascade of classifiers h, an initial structuring disk element of size D_I, a final disk size D_F, and a disk increment i.

$R \cup 0$
Compute the BSM and DTW features of all the set of positive examples P and the negative examples N.
for each structuring element D of size D_I, increasing by i, to D_F **do**
  $ImDilated$ = dilation of I using the disk D
  for each connected component r in $ImDilated$ of accepted size and area **do**
    test cascade h over region r

$$h(r) \quad = \quad \begin{cases} 1 & \text{if target detection, save region } R = R \cup r \\ 0 & \text{if background classification} \end{cases}$$

  end for
end for
Remove from R the repeated instances of a same clef.

Ensure: Target symbol regions R

In this way, only those regions that arrive to the last stage of the cascade are classified as clefs, are then selected as clef regions, and the rest of the regions are rejected. Each stage analyzes only the candidates accepted by the previous stages, and thus, the non-clefs are analyzed only until they are rejected by a stage. Notice that the BSM descriptor is used for the first classifier, because it is very fast to compute. It must be said that when dilating the image with different disk sizes, it may occur that several instances of a same clef have been accepted. In this cases, only one instance of each clef is stored.

5.4 Classification of Clefs

Once we have the clefs extracted from each music sheet of the database, the classification in terms of the writer is performed. It can be seen as a multi-class clef classification, in which all the music clefs detected from each page, must be assigned to the same writer. We propose a non-supervised approach, avoiding the definition of the clef for each writer in the database. Thus, the idea is to compare the detected clefs of the test music page with the clefs of the training database. For this purpose, the BSM descriptors previously computed are used to compute the distance between each clef (using the Euclidean distance and the k-NN classifier). The BSM features have been chosen (instead of the DTW-based features) because the segmented clefs in the symbol detection step usually have important noise and gaps (the DTW-based features are more sensible than the BSM features to this kind of distortions).

Then, the combination of the classification results of all the clefs (belonging the same music sheet) is performed so that each clef gives votes to the class of its nearest neighbor clefs of the training. This process has the following steps. First, each test clef is compared to the clefs of the training set using the k-NN classifier. For each clef, a list of the k nearest neighbor clefs is obtained, and sorted so that the first

candidate is the nearest neighbour of all. Then, the first ranked clef adds k votes to its corresponding class, the second nearest neighbor clef gives $k-1$ votes to its class, and this process is repeated until the last candidate adds one vote to its corresponding class. After the voting performed for each clef belonging to the music page, the test music score will be classified as the class which has received the maximum number of votes.

It must be said that if an input clef has no nearest neighbors in the database (the distance to all the BSM descriptors is higher than the value set in the training step), then, it is discarded, and consequently, it can not vote. In this way, the symbols that could be wrongly accepted as clefs (false positives), could be detected, and consequently, rejected from the voting stage.

5.5 Results

We have tested our method in a data set composed of 160 music sheets. They have been obtained from a collection of music scores of the 17th, 18th and 19th centuries, from two archives in Catalonia (Spain): the archive of Seminar of Barcelona and the archive of Canet de Mar. The data set contains 10 pages for each one of 16 different writers. Although we have performed a database of 10 pages for each one of the 20 different writers (which is described in the Appendix), we have decided to discard 4 writers for this experiment, because they contain music sheets without any clef (see Fig.5.2) or because they use to write only one clef for each page (not being enough for a good classification).

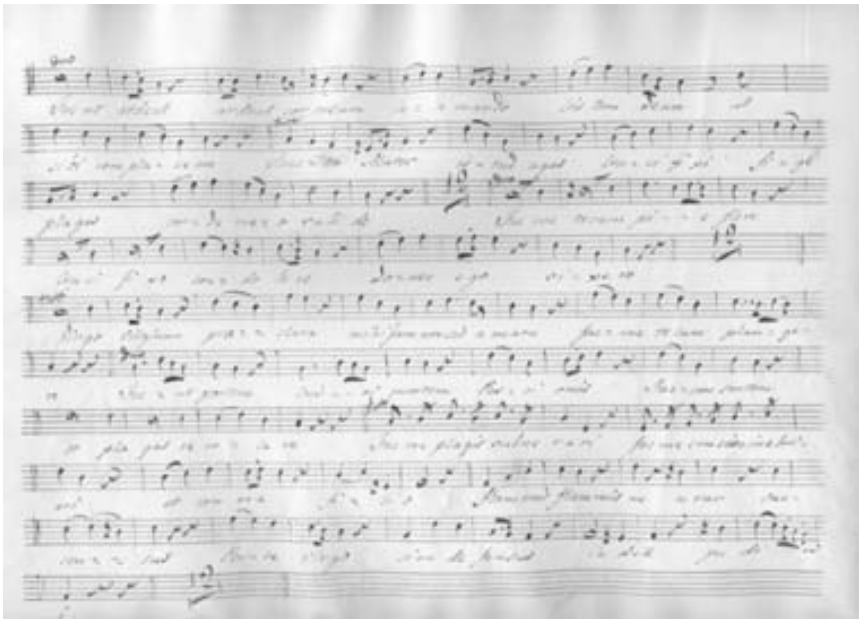


Figure 5.2: Example of an old score without any music clef.

After the preprocessing of each music sheet (in which it is binarized, deskewed and staff lines are removed), the symbol detection technique above described has been applied to extract the music clefs. In the learning stage of the detection process, the parameters for the BSM and the DTW-based method have been trained. As a result, the grid size for the BSM descriptor has been set to 25, and 7 features are used (the upper and lower profile, and 5 zones) for the DTW-based method.

The results of the clef-detection method applied to this database are shown in Table 5.2. The retrieved value corresponds to the number of symbols that have been detected as clefs using the symbol detection method. The true positive value indicates the number of retrieved clefs that are real clefs, whereas the false positive value indicates the wrongly detected clefs. Finally, the false negative value indicates the number of clefs that are missed. The database has 160 music sheets with a total of 733 music clefs. The method has correctly detected 592 clefs, has missed 141 clefs (false negatives), and has wrongly detected 697 regions as clefs (true positives). Thus, the detection rate is 81.6% (598/733), the false positive rate is 54% (697/1292) and the false negative rate is 19.2% (141/733). Having a look at this results, we can affirm that although the detection rate is acceptable, there is an important rate of missed clefs and false positives.

Concerning the false positive rate (over the 50%), in the classification step, the most part of these false negatives will find no nearest neighbor, and will not be allowed to participate in the voting. After examining the missed clefs, we can see that most of the missed clefs are the result of a bad segmentation, with important noise and gaps. These segmentation problems are due to the binarization and staff removal stages applied to documents with an important degree of degradation.

As an example, Figure5.3(a) shows two badly segmented clefs (with gaps) and the corresponding manually segmented clef (Fig.5.3(b)); and Figure.5.3(c) shows two badly segmented clefs (with noise from the staff lines) and the corresponding manually segmented clef (Fig.5.3(d)).

(a) (b) (c) (d)

Figure 5.3: Examples of segmented clefs. (a) Segmented treble clefs with gaps and the corresponding ideal segmented clef (b). (c) Segmented noisy bass clefs and the corresponding ideal segmented clef (d).

Concerning the writer identification experiments, we use 5-fold cross validation. We have used 5 test subsets, randomly chosen, containing one page per writer. For the BSM descriptor, the grid size with value 25 has been used. For each test subset or 16 images, the remaining 144 images are used for training. The classification has been

Writer	# Positives	# Retrieved	# True Pos.	# False Pos.	# False Neg.
1	16	36	15	21	1
2	73	145	70	75	3
3	63	78	42	36	21
4	70	87	54	33	16
5	34	29	22	4	12
6	65	68	56	12	9
7	70	96	65	32	5
8	31	37	22	15	9
9	57	70	53	17	4
10	23	83	14	69	9
11	67	103	51	52	16
12	55	114	36	78	19
13	45	79	43	36	2
14	13	47	12	35	1
15	15	54	14	39	1
16	36	166	23	143	13
Total	733	1292	592	697	141

Table 5.1: Symbol Detection Results: For each writer, the number # of retrieved regions, true positives, false positives and false negatives are shown.

performed using a k-Nearest Neighbor (k-NN) classifier based on Euclidean distance and cross validation, with the voting step previously described. We have used $k = 3$ and $k = 5$ for the k-NN, obtaining the same results. Table 5.2 shows the writer identification rates of each writer and for each set. Notice that each set contains only 1 page per writer, so the identification values can only be 0% or 100%. One can see that there are 12 writers with a 100% of identification rate, two writers with one wrong-classified pages, and 2 writers have two wrong-classified pages.

As a summary, the overall writer identification rate is 92.5% with these 16 writers. Although the database has a low number of writers, and the detection and segmentation of clefs should be improved, results show that this method is very promising.

5.6 Conclusions

In this Chapter we have analyzed and discussed the characteristic properties of the handwriting style for music notation, concluding that the music clef is a very good choice for discriminating the different writers. Afterwards, we have proposed a *symbol-dependent* writer identification method based on the shape or music clefs. It has been performed using a cascade of two classifiers for saving computational cost time. The classifiers are based on the computation of the BSM descriptor and the DTW-based features, which are fully described in the previous Chapters. After detecting and segmenting the clefs, the classification is performed using a non-supervised approach, in which the clefs belonging to the test music pages are compared to the clefs from the training music sheets.

Writer	Set 1	Set 2	Set 3	Set 4	Set 5	**Average**
1	100%	100%	100%	100%	100%	100%
2	**0%**	**0%**	100%	100%	100%	60%
3	100%	100%	100%	100%	100%	100%
4	100%	100%	100%	100%	100%	100%
5	100%	100%	100%	100%	**0%**	80%
6	100%	100%	100%	100%	100%	100%
7	100%	100%	100%	100%	100%	100%
8	100%	100%	100%	**0%**	100%	80%
9	100%	100%	100%	100%	100%	100%
10	100%	100%	100%	100%	100%	100%
11	100%	100%	100%	100%	100%	100%
12	100%	100%	100%	100%	100%	100%
13	100%	100%	100%	100%	100%	100%
14	100%	100%	100%	100%	100%	100%
15	100%	**0%**	100%	**0%**	100%	60%
16	100%	100%	100%	100%	100%	100%
Overall	93.75%	87.5%	100%	87.5%	93.75%	**92.5%**

Table 5.2: Classification Results: Writer identification rates for the 16 writers.

Concerning the clef-detection technique, results show that although there is an important amount of false positives, and a small set of false negatives, the retrieval of clefs is enough accurate for the writer identification method. Results show the good writer identification rate (92%) in a database of 16 writers and 160 music pages. Although the method should be applied to a bigger database (in fact, some writers have been discarded because of the limitations of these small database), the promising results show that this method has a very high discriminatory power.

As a summary, we can affirm that the method is very promising. It must be said that the performance of the proposed writer identification method is closely related to the performance of the detection and segmentation of clefs. Thus, a more accurate symbol-detection technique, will obviously increase the final writer identification rate.

Chapter 6

A Symbol-Independent Writer Identification Approach based on Features of Music Lines

The aim of writer identification is determining the writer of a piece of handwriting from a set of writers. Contrary to the approach proposed in the previous Chapter, we present here a *symbol-independent* approach for writer identification in old handwritten music scores. The steps of the proposed system are the following. The music sheet is preprocessed and normalized for obtaining single binarized music lines, without the staff lines. Afterwards, 98 features are extracted for every music line, which are subsequently used in a k-NN classifier that compares every feature vector with prototypes stored in a database. The proposed method has been tested on a database of old music scores from the 17th to 19th centuries, achieving encouraging identification rates.

6.1 Introduction

Writer identification is focused on the identification of the author of a piece of handwriting from 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, the meaning of the text is known, whereas in the second one, the writer identification can be performed without recognizing any words. One of the aims of this work consists in extending the second approach to music scores, obtaining a *symbol-independent* method.

As it has been commented in the Introduction, the identification of the author of a handwritten document in terms of graphical information is still a challenge. In the previous Chapter, a *symbol-dependent* method has been proposed, which detects and recognizes the music clefs and performs writer identification in music scores based on the shape descriptors computed from each clef. As it has been concluded in the previous Chapter, the performance of the method is related to the performance of the

detection and segmentation of clefs. The more accuracy in the clef-spotting technique, the higher performance of the writer identification method.

In this Chapter and in the following one, we study the adaptability of some existing writer identification approaches for text documents to music scores. The objective is to adapt two approaches which have successfully been used for performing writer identification in text documents. The first one consists in using features extracted from text lines [HB03], and the second one consists in extracting features from texture images [STB00].

In this Chapter we present the first off-line symbol-independent proposal for performing writer identification in musical scores, which avoids the recognition of the elements in the score. Some authors (see [STB00], [BS06], [SB08]) claim that writer identification in handwritten text documents can be performed without recognizing any words, i.e., with the meaning of the text being unknown. In the present Chapter, this assumption is extended to music scores. Consequently the system will be faster and more robust, avoiding the dependence on a good music recognizer. In fact, we have adapted part of the writer identification approach described by Hertel and Bunke in [HB03] to old musical scores, where instead of letters of the alphabet, music symbols are analysed.

The remainder of this Chapter is structured as follows. In the next section the preprocessing steps are presented, in which the music score is binarized, staffs are remove and the music line is normalized. In Section 3 the feature extraction approach is described, in which 98 features are computed from basic measures (such as slant, width of the writing), compounding primitives, contours and fractals. In Section 4 some feature set search methods are described. Experimental results are presented in Section 5. Finally, Section 6 concludes the Chapter.

6.2 Preprocessing

The preprocessing phase consists in binarizing the image, removing staff lines and normalizing the musical lines. Every output file contains the musical notation of one staff line. The process is described in the following subsections.

6.2.1 Binarization and Staff removal

The input gray-level scanned image (at a resolution of 300 dpi) is first binarized with the adaptive binarization technique proposed by Niblack [Nib86]. 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 method is used to detect lines and obtain the orientation of the music sheet. Then the image is rotated if necessary.

For 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 (even if they are printed) is difficult 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 are only 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 above described is consulted. Finally, those segments that belong to the staff lines (their width is similar to the average of the width of staff lines, which has been computed previously) are removed. For further details, see Chapter 8, in which the application scenario is described.

6.2.2 Normalization

The information about 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 upper or lower in the staff for reaching higher or lower frequency. Therefore, melodies with both treble and bass notes will 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. For that reason, the music notes must be 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 an horizontal reference line (see Fig.6.1).

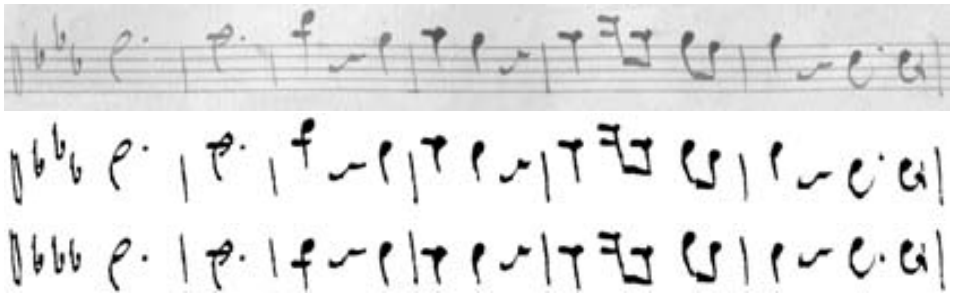


Figure 6.1: Preprocessing step: Original music line in gray scale, binarized music line (without staff lines), and normalized line, in which all the music symbols are aligned in respect to a horizontal line.

For the obtention of the music line that will be used for the computation of features, our first option consisted in generating as many music lines as staffs are drawn in the music sheet. Thus, each music staff line was preprocessed, normalized and

stored as a 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 classifier, because it has been noticed that some music sheets contain short staffs. In this cases, each music line do not contain enough music symbols to compute reliable features, because small differences usually create outlier values as features, and will consequently confuse the classifier.

For avoiding this problem, each music page will generate exactly three music lines, independently of the number of music staff lines that it contains. We have generated three lines after revising the amount of music symbols that usually appear in the music sheets. 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 features extraction stage (see Fig.6.2).

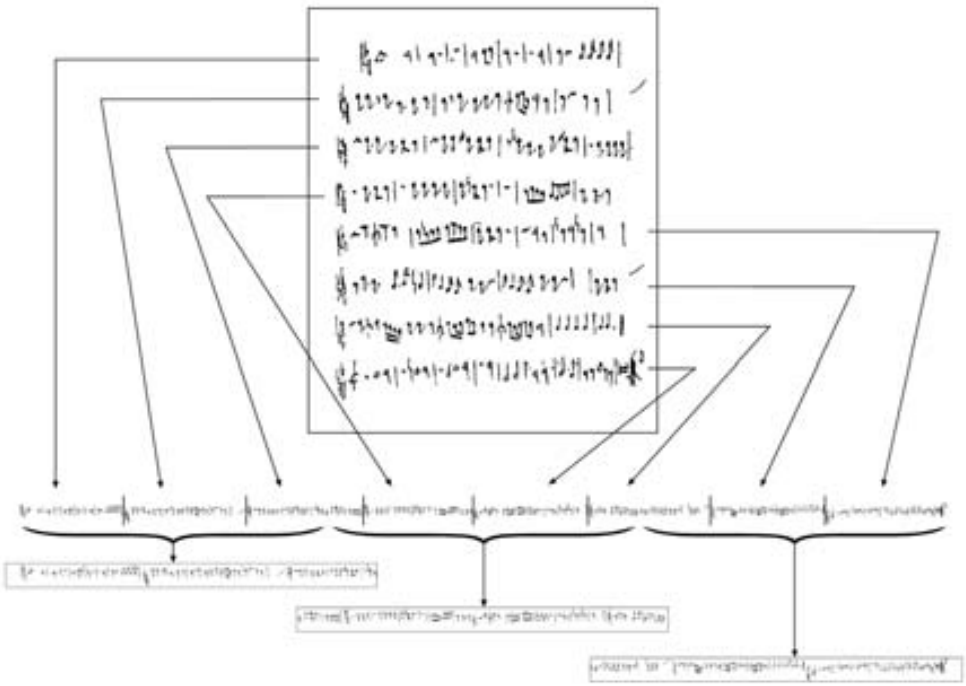


Figure 6.2: Obtention of the three music lines for each page. Once the staff lines are removed, each music line is normalized and joined in a single music line. Then, this line is split in three lines which will be stored as the input music lines.

6.3 Feature Extraction

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 [HB03] 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 deleted in our approach are the enclosed 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. For this reason, the shape of the loops is useful, thus the two features are consequently added to the set of features defined for writer identification in text documents. Contrary, the probability of finding closed loops in music notation is low. In fact, just a few number of music symbols contain loops (e.g. whole and half note or accidentals), and in addition, they are not frequent. Consequently, these symbols appear only in a small subset of music lines, and for this reason, they can not be used for writer identification in music scores.

A brief description of the features extracted is given below. For a full description we refer to [HB03] and [MMB01].

6.3.1 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 histogram 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 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 projection. 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.

The width of the writing is obtained by selecting the row with most black-white and white-black transitions. Here, and for avoiding outliers, the median m_l of the lengths of every run 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.

6.3.2 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 elements belonging to the same connected component. Moreover, the average, median, standard deviation of the length of the connected components are used as features.

6.3.3 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, being useful information 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. Once the continuous lower and upper contour (called characteristic contours) are obtained, the following features are extracted: slant 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 the same for local minima to the right and to the left.

6.3.4 Fractal Features

The idea proposed in [BVSE97],[BVSE98] 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 this kernels, the area $A(X_\eta)$ of the dilated writing X_η is measured. The fractal dimension $D(X)$ is defined by:

$$D(X) = \lim_{\eta \to 0} (2 - \frac{\ln A(X_\eta)}{\ln \eta}) \tag{6.1}$$

Then, we obtain the evolution graph plotting the behaviour of y over x (see Fig.6.3):

$$x = \ln \eta; \qquad y = \ln A(X_\eta) - \ln \eta \tag{6.2}$$

Afterwards, this function is approximated by three straight lines (see Fig.6.3). The points $p_1, ..., p_4$ 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 information about the rotation in the writting style. These ellipses are defined with increasing the length of the ellipse's two main axes and the rotation angle. Thus, a total of 63 (=21x3) features are extracted.

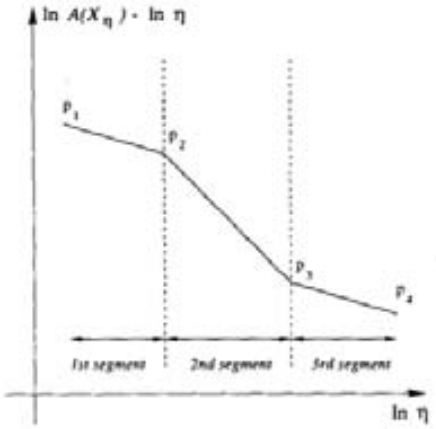


Figure 6.3: Fractals: Approximation of the evolution graph by three straight lines (extracted from [MMB01]).

6.4 Feature Selection

In [SKB05] the suitability of the 100 Hertel's 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 original ones, and also, results in a more efficient classifier.

One very well know technique is the Feature Set Search, which looks for the best subset of features for classification. In [Kit78] and [PJJ94] four Feature Set Searching techniques are described: Sequential Forward Search (SFS), Sequential Backward Search (SBS), Sequential Floating Forward Search (SFFS), Sequential Floating Backward Search (SFBS).

SFS starts with an empty set of features, and at each step one single feature is added to the set. The feature chosen is the best classifying feature from the remaining set of features. Contrary, SBS starts with the full set of features, and removes one feature so that the new reduced set of features yields a higher writer identification rate. SFBS and SFFS are an improvement of SFS and SBS, adding the ability to do backtracking. The set of features can be incremented or reduced by one feature at each time, changing dynamically the number of features in the set, thus floating up and down. SFBS starts with the empty set of features, whereas SBS starts with the full set of features.

6.5 Experimental Results

We have tested our method in a data set consisting of 200 music sheets. They have been obtained from a collection of music scores of the 17th, 18th and 19th centuries, from two archives in Catalonia (Spain): the archive of Seminar of Barcelona and the archive of Canet de Mar. This database is fully described in the Appendix. An example of an old score can be seen in Figure 6.4. The data set contains 10 pages for each one of 20 different writers. For each page, we generate 3 music lines, obtaining a database of a total of 600 music lines (3 music lines x 10 pages x 20 writers). The music lines are obtained through the preprocessing steps described above, and the vector of 98 features is computed for every music line.

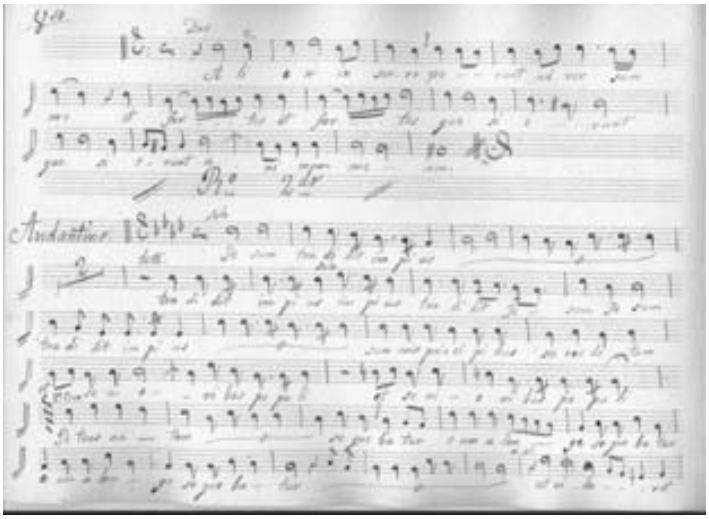


Figure 6.4: Example of an old score of the composer Casanoves.

For the experiments, we have used 5 test subsets, randomly chosen, containing one page per writer. This means that all the three music lines obtained from every page are used in the test set. Due to the importance of the obtention of independent test subsets, all the three lines obtained from one music page belong to the same subset. For each test subset or 60 images, the remaining 540 images are used for training. The classification has been performed using a k-Nearest Neighbor (k-NN) classifier based on Euclidean distance and cross validation.

As it has been said, the three music lines that belong to the same music sheet should be only assigned to one class. Concerning the combination of the classification results of these three lines, in the experimental results, the Majority Voting and Borda Count combination methods are compared to the base one, in which no combination is performed, and for each page, the three music lines could be assigned to different writers. The reader is referred to Chapter 8 (the Application scenario), in which these combination methods are explained.

In Table 6.1 the writer identification results for an increasing set of writers are shown. From the database, the first 5 writers have been selected, and the classification

rates have been obtained for different values of k-NN and different combination 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 seen that 3-NN and 5-NN obtain in most cases the better recognition rates. Results using Majority Voting or Borda Count are better than results using no combination at all. Concerning the scalability of the method, in the best cases, the recognition rate of 84% for 5 writers decreases to 76% for 20 writers, showing that the method has a good scalability degree.

W.I.Rate	Combination	1-NN	3-NN	5-NN	7-NN	9-NN
5 writers	None	77.3%	77.3%	77.3%	73.3%	74.6%
5 writers	Majority Voting	76%	84%	76%	72%	76%
5 writers	Borda Count	76%	**84%**	**84%**	72%	72%
10 writers	None	75.9%	75.3%	72%	68%	66%
10 writers	Majority Voting	**82%**	80%	74%	68%	70%
10 writers	Borda Count	**82%**	80%	76%	72%	70%
15 writers	None	71.1%	69.3%	69.7%	69.3%	68.8%
15 writers	Majority Voting	76%	**78.6%**	74.6%	70.6%	73.3%
15 writers	Borda Count	76%	77.3%	73.3%	73.3%	73.3%
20 writers	None	69.6%	70.6%	68.3%	68.6%	68.6%
20 writers	Majority Voting	74%	**76%**	74%	71%	73%
20 writers	Borda Count	74%	75%	75%	73%	73%

Table 6.1: Classification Results: Writer identification rates using 98 line features for different database sizes and different combination of results.

Concerning the SFS, SBS, SFFS and SFBS experiments, wrappers are used as objective function, where one of the five subsets is used as the test set and the others as prototypes in the 5-NN classifier. To evaluate the fitness of a selected feature subset, iteratively three subsets are used in the classifier and the remaining set is used to measure the fitness of the feature subset under consideration. Once the algorithm finds the best feature subset, the fifth subset is used for the final writer identification rate. In Table 6.2 results of feature selection algorithms are shown 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.

It must be said that, although they do not reach any improvement over the baseline (SFBS reaches 75% which is similar than 76% of the baseline rate), the dimensionality reduction is significant (from the 98 features of the baseline to the 20 features selected with 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. This is related to the course of dimensionality, showing that in classification domains, the number of features is not related to the final classification rate, because there is a moment in which the increasing number of features (and consequently, an increasing of the dimension), instead of helping in the classification, they introduce noise and more confusion in the classes.

W.I.Rate	Combination	N. of Features	3-NN	5-NN
All Features	None	98	70.6%	68.3%
All Features	Majority Voting	98	**76%**	74%
All Features	Borda Count	98	75%	75%
SFS	None	43	60%	58.3%
SFS	Majority Voting	43	65%	60%
SFS	Borda Count	43	65%	60%
SBS	None	54	65%	58.3%
SBS	Majority Voting	54	60%	65%
SBS	Borda Count	54	65%	60%
SFFS	None	35	65%	66.6%
SFFS	Majority Voting	35	70%	70%
SFFS	Borda Count	35	70%	70%
SFBS	None	20	66.6%	68.3%
SFBS	Majority Voting	20	70%	**75%**
SFBS	Borda Count	20	**75%**	**75%**

Table 6.2: Classification Results: Writer identification rates for 20 writers using Feature Set Search methods.

6.6 Conclusions

In this Chapter we have presented a *symbol-independent* method for writer identification in musical scores. The steps of the system are the following. In the preprocessing step, the image is binarized, de-skewed, staffs and removed and the lines of music symbols are normalized. Afterwards, 98 features (slant, connected components, upper and lower contours, and fractals) are computed. Finally, the classification is performed using the k-Nearest Neighbour method, and several combinations of results, so that all the music lines belonging to the same music sheet are classified to the same class.

Experimental results show that the method has obtained promising results, with a good scalability degree. Concerning the combination methods, the Borda Count obtains a identification rate of 75%, whereas the Majority Voting obtains a identification rate of 76%. This results show that there is not a significant improvement, and both combination methods can be used. In the same way, there is no significant improvement between the 3-NN and 5-NN classification method, being both values suitable for this method. Although Feature Set Search methods show that some of the 98 features are redundant or irrelevant. It must be said that these selected features are specific to this database, and the results could potentially be quite different for other datasets. In the same way, the use of other feature set search of combination methods could obtain different results.

A second *symbol-independent* writer identification method will be described in next Chapter, based on the extraction of features from music texture images.

Chapter 7

A Symbol-Independent Writer Identification Approach Based on Features from Texture Images

As a continuation of the previous Chapter, we present another blind-approach for writer identification in old handwritten music scores. It is an adaptation of the textural approach used for writer identification in text documents described by Said[STB00]. The steps of the proposed system are the following. First of all, the music sheet is preprocessed for obtaining a music score without the staff lines. Afterwards, five different methods for generating texture images from music symbols are applied. Every approach uses a different spatial variation when combining the music symbols to generate the textures. Finally, Gabor filters and Grey-scale Co-ocurrence matrices are used to obtain the features. The classification is performed using a k-NN classifier based on Euclidean distance. The proposed method has been tested on a database of old music scores, achieving promising identification rates.

7.1 Introduction

In the previous Chapter we presented an approach for writer identification using 98 features extracted from music lines. Those features were derived from connected components, contours, fractals and basic measurements. The experimental results using those local features were quite good, but in some cases a single music line has not enough information to identify the writer correctly. In this Chapter we propose the use of textural features, because they are able to represent the music score globally rather than focusing on a set of predefined local features.

Textures provide important characteristics for object identification, playing an important role in image analysis and pattern classification [TJ98], [Har79]. Texture classification has been used in applications such as biomedical image processing, content based image retrieval, the analysis of satellite images, etc. As it has been

commented in Chapter 2, some authors ([STB00], [GBA07], [HS08]) treat writer identification as a texture identification problem. They first generate a uniform texture from text lines, and then, they compute textural features. Some works ([ZTW01], [HS08]) demonstrate that these features can been also used for script and language identification. Concretely, Peake and Tan [PT97] propose the generation of texture images from printed text for script and language identification. The method uses Gabor filters and grey level co-ocurrence matrices as textural features, and classifies with the k-NN classifier. In [STB00], texture images are generated from handwritten text for writer identification. These approaches demonstrate that textural features can be successfully used for writer and script identification.

In the current Chapter we have adapted part of the writer identification approach described by Said et al. in [STB00] to old musical scores, where instead of words, music symbols are used for generating textures, and consequently, textural features can be computed for the identification of the writer. After the preprocessing of the music sheet, five different approaches have been applied for the generation of image textures with music symbols. Once we have the music textures, textural features can be computed. In principle, any textural feature can be applied to texture images. In this work we propose the computation of Gabor filters and Gray-Scale co-ocurrence matrices (GSCM). Finally, the classification is performed using the k-NN classifier, and some feature selection methods are used to increase the writer identification rates.

The remainder of the Chapter is structured as follows. In the next section the preprocessing and the generation of textures are presented, and in Section 3 the feature extraction approach is fully described. Experimental results are presented and discussed in Section 4. Finally, Section 5 concludes the Chapter.

7.2 Preprocessing and Generation of Textures

The preprocessing phase consists in binarizing the image, removing staff lines and generating the texture images from music notes. The process is described next.

7.2.1 Binarization and Staff Removal

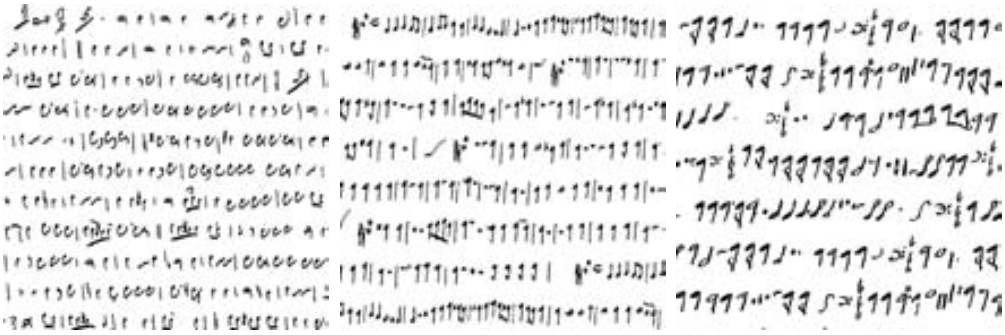
First of all, the input gray-level scanned image (at a resolution of 300 dpi) is binarized (with the adaptive binarization technique proposed by Niblack [Nib86]), and filtering and morphological operations are applied to reduce noise. Then, the image is deskewed using the Hough Transform. Afterwards, the staff lines are removed, because they are usually printed, and consequently, they are not useful for writer identification. The staff removal process must cope with paper degradation, the warping effect, distortions and gaps. The method proposed consists in obtaining a coarse approximation of the staff lines applying median filters with a horizontal mask and then reconstructing the staff joining these segments. Afterwards, a contour tracking process is used for following and removing every staff line, taking into account the coarse approximation when gaps are appearing. For further details, see Chapter 8, where the complete application is described.

7.2.2 Generation of Music Textures

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

We have applied five different methods for obtaining the texture images. Each method is characterized by a different spatial variation when combining the music symbols to generate the textures. In all cases, the size of the texture image is of 2048x2048 pixels. The following five different methods for obtaining the textures have been applied:

1. Basic Texture: It consists in taking all the music symbols obtained after the staff removal step, without any other processing (see Fig. 7.1). In this way, the music symbols appear in the same order than in the music score, keeping the inter-symbol distance. It can be seen as a squarish piece of handwriting extracted from the music sheet.



(a) (b) (c)

Figure 7.1: Basic texture images generated from music lines of three different writers. (a) Writer 1, (b) Writer 2, (c) Writer 3.

2. TextLine Texture: It consists in taking randomly music symbols and putting them in a reference line, with the same inter-symbol distance (see Fig. 7.2). In this way, if the music score contains a group of the same kind of music symbol (i.e. quarters or rests), they will be randomly distributed over the texture, achieving texture independence of the rhythm. In addition, the image texture will contain more symbols, resulting in a more dense texture.

3. Random Texture: It consists in taking randomly music symbols and putting them in random locations of the image, (see Fig. 7.3). In this way, not only the music symbols are randomly chosen, but also they are randomly distributed along the image. In this way, the high frequencies (the horizontal distribution of symbols) that interfere in the representation space of the Basic and TextLine textures are avoided.

(a) (b) (c)

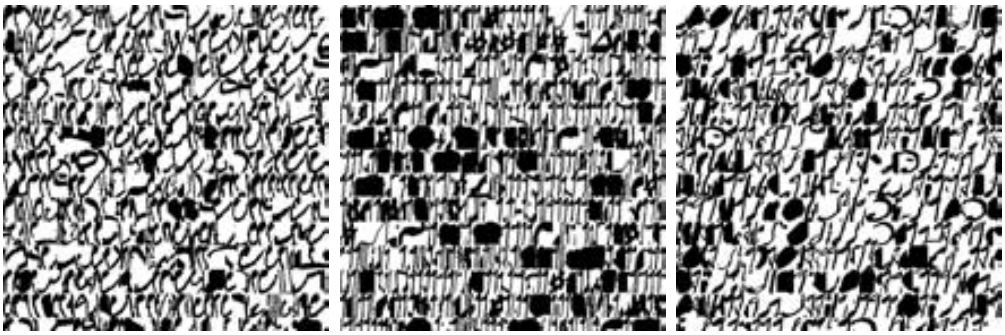
Figure 7.2: TextLine texture images generated from music lines of three different writers. (a) Writer 1, (b) Writer 2, (c) Writer 3.

(a) (b) (c)

Figure 7.3: Random texture images generated from music lines of three different writers. (a) Writer 1, (b) Writer 2, (c) Writer 3.

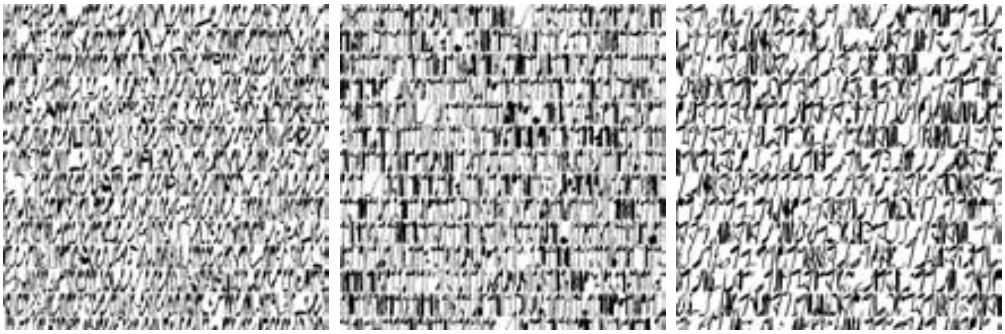
4. AspectRatio Texture: It consists in taking the idea of TextLine texture, but making all the symbols of equal size (see Fig. 7.4). For every symbol that must be resized, its aspect ratio will be maintained. The main purpose is to avoid gaps in the texture, obtaining a higher density of the texture.

5. Resize Texture: It consists in the same idea as AspectRatio Texture, but without the preservation of the aspect ratio in the resizing process (see Fig. 7.5). In this way, the appearance of the symbol is distorted (symbols are taller comparing to the original shape).



(a) (b) (c)

Figure 7.4: AspectRatio texture images generated from music lines of three different writers. (a) Writer 1, (b) Writer 2, (c) Writer 3.



(a) (b) (c)

Figure 7.5: Resize texture images generated from music lines of three different writers. (a) Writer 1, (b) Writer 2, (c) Writer 3.

Notice that the first two approaches generate texture images that look like a music score, whereas the last three approaches generate more compact and synthetic texture images. In fact, the AspectRatio texture has the inconvenient of creating some big black areas, because small compact music symbols (such as dots and half rests) are extremely enlarged. It is important to remark that in all texture images, the three writers can be easily distinguished one from each other. Having a look at the resulting texture images (see Figures 7.1, 7.2, 7.3, 7.4, 7.5), one can see that the 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 an important slant degree.

7.3 Feature Extraction from Textures

Once we have the images of music textures, textural features can be computed. In [STB00] and [PT97], 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 for being able to extract textural features.

Next we describe the textural features which are computed in our approach: Gabor features and Gray-Scale co-ocurrence matrices.

7.3.1 Gabor Features

The multi-channel Gabor filtering technique [Tan92] can be seen as a window Fourier Transform in which the window function is Gaussian. This technique is based on the psycophysical findings that affirm that the processing of pictorial 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 are of opposite symmetry and are computed as:

$$\begin{cases} h_e(x, y; f, \theta) = g(x, y)cos(2\pi f(xcos\theta + ysin\theta)) \\ h_o(x, y; f, \theta) = g(x, y)sin(2\pi f(xcos\theta + ysin\theta)) \end{cases} \tag{7.1}$$

where $g(x, y)$ is a 2D Gaussian function, the central frequency is f, and θ corresponds to the orientation which define the location of the channel in the frequency plane. Afterwards, the Fourier transform (FFT) of the filters are computed as:

$$\begin{cases} q_e(x, y) = FFT^{-1}\left[P(u, v)H_e(u, v)\right] \\ q_o(x, y) = FFT^{-1}\left[P(u, v)H_o(u, v)\right] \end{cases} \tag{7.2}$$

where $P(u, v)$ 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. Finally, we perform a combination of the two filters, and a single value at each pixel is obtained:

$$q(x, y) = \sqrt{q_e^2(x, y) + q_o^2(x, y)} \tag{7.3}$$

For the computation of features, we have to define the angle θ and the central frequency f, which specify the location of the Gabor filter on the frequency plane. In [Tan96], it has been shown that for an image of size NxN, 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 $f = \{4, 8, 16, 32\}$ and the orientation with values $\theta = \{0^o, 45^o, 90^o, 135^o\}$. The output corresponds to $4x4 = 16$ images. Extracting the mean and the standard deviation we obtain a total of $16x2 = 32$ features.

7.3.2 GSCM features

Some authors [TJ98], [HSD73] maintain that the neighbourhood properties can also represent a texture. In this sense, the grey level co-occurrence and their distribution in the pixel neighbourhood reflect the local activities of a texture, being one of the useful neighbourhood properties used for texture description. They estimate image properties related to second-order statistics, allowing the discrimination of one texture from another. Haralick [HSD73] proposes the use of Grey-Scale Co-ocurrence Matrices (GSCM), which describe the pair of grey levels with special distance and special 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 (see [PT97]).

If an image contains N grey levels, for every distance d and angle θ we obtain a matrix NxN defined as $GSCM_{d,\theta}$, where $GSCM_{d,\theta}(a,b)$ corresponds to the number of pairs $(P1, P2)$ where $P1$ is of grey value a, $P2$ is 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 $d = \{1, 2, 3, 4, 5\}$; and the orientation $\theta = \{0^o, 45^o, 90^o, 135^o\}$. The output corresponds to 20 matrices of dimension 2x2, and due to the diagonal symmetry, there are only 3 independent values in each matrix. In total we obtain 20x3 = 60 features.

7.4 Experimental Results

We have tested our method with 200 music pages from 20 different writers, where every writer has written 10 pages. The music pages coincide with the ones used for evaluating the writer identification method based on line features, and are fully described in the Appendix. Due to the large amount of symbols on every music page, three different texture images can be generated for each music page, obtaining a database of 20x10x3 = 600 music textures.

For the experiments, we have used 5 test subsets, randomly chosen, containing one page per writer. This means that all the three music textures obtained from every page are used in the test set. Due to the importance of the obtention of independent test subsets, all the three textures generated from one music page belong to the same subset. For each test subset or 60 images, the remaining 540 images are used for training. The classification has been performed using a k-Nearest Neighbor classifier based on Euclidean distance and cross validation. Due to the fact that every music page generates three texture images, the three texture images should be assigned to only one class. For this reason, the experiments show the no combination of results (the three textures of a same page could be assigned to different classes) and also the combination of the classification results using the Majority Voting and Borda Count methods. These combination methods have been described in Chapter 8.

In Table 7.1 the writer identification rates (w.i.r.) for the Basic, Textline, Random, AspectRatio and Resize textures are shown. We can see the results for different values of $k = 3, 5, 7$ for the Nearest Neighbor, and also with the Borda Count method for

the combination of classification results. Notice that in most of the cases, the Gabor features obtain lower identification rates than the GSCM features (except for the AspRatio textures). In addition, the combination of the Gabor and GSCM features in one single vector (of 92 features) increases the final recognition rates only for the Random and Resize textures. One can see that the Borda Count method usually increases the final classification results (excepts in the TextLine textures), and in most of the cases, the $k = 5$ value slightly increases the final classification rates.

Concerning the texture images used, Resize textures obtain the highest w.i.r. in all the cases, reaching a 73% of w.i.r. using the combination of Gabor and GSCM features. Random, Basic and AspectRatio textures reach lower identification rates (58% of the Basic textures with GSCM, 59% with Random Features using GSCM and Gabor features, and the 65% of AspRatio textures using Gabor features). Contrary, the textures extracted using the TextLine method obtain in all cases the lowest rates (under 50% or w.i.r).

Features	#	Combin.	Basic	TextLine	Random	A.Ratio	Resize
Gabor 3-NN	32	None	46%	34%	44%	54%	59%
Gabor 5-NN	32	None	44%	34%	42%	56%	59%
Gabor 7-NN	32	None	45%	33%	44%	55%	62%
Gabor 3-NN	32	B. Count	53%	34%	54%	62%	67%
Gabor 5-NN	32	B. Count	53%	34%	54%	**65%**	64%
Gabor 7-NN	32	B. Count	54%	34%	53%	61%	66%
GSCM 3-NN	60	None	54%	47%	48%	46%	64%
GSCM 5-NN	60	None	53%	48%	48%	47%	64%
GSCM 7-NN	60	None	52%	**49%**	48%	46%	61%
GSCM 3-NN	60	B. Count	55%	45%	53%	58%	64%
GSCM 5-NN	60	B. Count	**58%**	45%	55%	55%	66%
GSCM 7-NN	60	B. Count	56%	46%	56%	53%	66%
Both 3-NN	92	None	54%	46%	50%	50%	67%
Both 5-NN	92	None	53%	45%	49%	53%	68%
Both 7-NN	92	None	52%	47%	47%	53%	68%
Both 3-NN	92	B. Count	54%	47%	53%	51%	70%
Both 5-NN	92	B. Count	55%	47%	**59%**	52%	**73%**
Both 7-NN	92	B. Count	56%	47%	59%	52%	71%

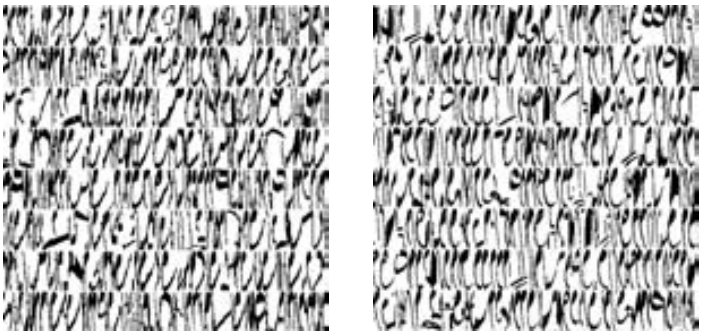
Table 7.1: Writer identification rates using Gabor and GSCM features for the five methods applied for obtaining texture images. It shows the results with the combination of features using the Borda Count method, or without any combination.

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 73% of writer identification rate using Borda Count and the 5-NN classifier. Although this kind of texture is performed by the distortion of music symbols, it is the most dense texture image of the all five options. 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. In the following experiments, we will only use the Resize Textures.

Concerning the scalability of the method, Table 7.2 shows the writer identification rates of Resize 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 confusion matrices analyzed show that the disciples of the same musician (or that belong to the same place and time period) tend to have a very similar writer style (see Figure 7.6).

W.I.Rate	Combination	3-NN	5-NN	7-NN
5 writers	None	93%	93%	89%
5 writers	Majority Voting	**96%**	92%	88%
5 writers	Borda Count	92%	92%	92%
10 writers	None	81%	81%	82%
10 writers	Majority Voting	**86%**	82%	80%
10 writers	Borda Count	84%	84%	82%
15 writers	None	67%	68%	69%
15 writers	Majority Voting	71%	**75%**	71%
15 writers	Borda Count	69%	73%	72%
20 writers	None	67%	68%	68%
20 writers	Majority Voting	73%	72%	71%
20 writers	Borda Count	70%	**73%**	71%

Table 7.2: Classification Results or Resize Textures: Writer identification rates using the 92 textural features (Gabor and GSCM) for different database sizes and different combination of results.



(a) (b)

Figure 7.6: Resize texture images from two writers: Both texture images are very similar although they belong to different classes.

7.4.1 Results Using Feature Selection Methods

The suitability of the textural features has also 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 original ones. As in the experiments of the previous Chapter, we have performed the Sequential Forward Search (SFS), Sequential Backward Search (SBS), Sequential Floating Forward Search (SFFS), Sequential Floating Backward Search (SFBS) (see [Kit78], [PJJ94]). 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 feature subset under consideration. Once the algorithm finds the best feature subset, the fifth subset is used for the final writer identification rate.

In Table 7.3 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. It is important to remark that they do not improve the identification results, although Majority Voting and Borda Count are used to increase the final identification rate. This fact shows that there are not many dependent or irrelevant features in the original feature set, being all the features important for the classification. Notice that these selected features are specific to this database, and the results could potentially be quite different for other datasets.

W.I.Rate	Combination	N. of Features	3-NN	5-NN
All Features	None	92	67%	68%
All Features	Majority Voting	92	73%	72%
All Features	Borda Count	92	70%	**73%**
SFS	None	32	61.6%	60%
SFS	Majority Voting	32	70%	60%
SFS	Borda Count	32	65%	70%
SBS	None	18	66.6%	63.3%
SBS	Majority Voting	18	65%	65%
SBS	Borda Count	18	65%	65%
SFFS	None	28	65%	66.6%
SFFS	Majority Voting	28	70%	65%
SFFS	Borda Count	28	70%	70%
SFBS	None	11	68.3%	**71.6%**
SFBS	Majority Voting	11	70%	70%
SFBS	Borda Count	11	70%	70%

Table 7.3: Classification Results: Writer identification rates for the 20 writers using Feature Set Search methods for Resize Textures.

7.5 Conclusions

In this Chapter we have presented another blind method for writer identification in musical scores using textural features. It is an adaptation of the text-idenpendent writer identification approach proposed in [STB00] to music scores. Consequently, the system is more robust, avoiding the dependence of a good recognizer. The steps of the system are the following. In the preprocessing step, the image is binarized, de-skewed, staffs are removed and the music textures are created. Afterwards, GSCM and Gabor features are computed, and the k-Nearest Neighbour rule is used for classification.

The experimental results show that some methods for generating textures are better than others. In fact, although Resize textures are the ones with the highest classification rates (even when the writer styles are very similar), the work could be extended if the textural features obtained from the five different approaches are combined in a single vector, so that the feature selection methods could possibly increase the final classification rate.

After describing the three different approaches for writer identification, in next Chapter we explain the combination of them, in order to improve the performance.

Chapter 8

Application Scenario on Writer Identification in Old Handwritten Music Scores

In this Chapter we describe the general ensemble architecture which combines the three writer identification proposed methods. First of all, an overview of the architecture is presented. Secondly, the preprocessing step applied for the three writer identification methods described, which consists in the binarization, deskewing and staff removal. Finally, in the post-processing step, the combination of the three identification methods is described. Results show that the combination of the three approaches significantly improve the final writer identification rate.

8.1 Introduction

In the previous Chapters, we have proposed three different methods (based on features from music lines, features from textures and symbol recognition features from clefs) for identifying the writer of a music score. They obtain quite good results, but a combination of the three methods is desired. In this Chapter, we propose an architecture for combining the identification results of the three approaches. As a result, the global identification rates can be improved.

In addition, the preprocessing step of the music score is described. It must be said that the three proposed approaches require a binarized image, without staff lines. As it has been commented in Chapter 1, there are some restrictions when working with old documents, because of paper degradation. Thus, there is an important problem of low level processing, because one must cope with the show-through and bleed-through problems, spots, stains, gaps and low contrast. For those reasons, it is necessary to use image enhancement processes to solve this kind of difficulties, but the research in this field is out of the scope of this thesis. A good option for dealing with paper degradation is the use of local binarization techniques, filtering and morphological

operations, whereas staff detection can be performed using a contour tracking process.

This Chapter is organized as follows. Section 2 presents and overview of the ensemble architecture, showing how the different approaches are combined. Section 3 describes the preprocessing stage, which is common to the three approaches. Section 4 describes the combination architecture. Experimental results are shown in Section 5. Finally, concluding remarks are exposed in Section 6.

8.2 Overview of the Ensemble Architecture

In order to combine the three writer identification approaches described in the previous Chapters, an ensemble architecture has been designed (see Fig.8.1). Firstly, the input image is preprocessed applying the preprocessing stage described in the next Section. The preprocessing consists in binarizing, deskewing and removing the staff lines and lyrics. The resulting image is then the input for the three writer identification approaches. The symbol-based method performs symbol spotting for detecting and extracting the music clefs from the image (see Chapter 5), the method based on music lines performs the specific preprocessing and normalitzation in order to obtain three music lines (see Chapter 6), and the method based on textures generates the three texture images (see Chapter 7).

The next step consists in computing the features for the lines, textures and clefs: the 98 line features are computed from the music lines, the Gabor filters and GSCM textural features are computed from the textural images, and the BSM descriptor is computed from each detected clef.

Once we have the extracted features for each approach, the post-processing is applied for the final classification. For this purpose, the combination of results is performed using the Borda Count method, so that each element (line, texture or clef) gives votes to the nearest neighbor classes. Finally, the votes of the three approaches are taken into account for the final identification of the input image. The reader is referred to the post-processing Section for further details about this last stage.

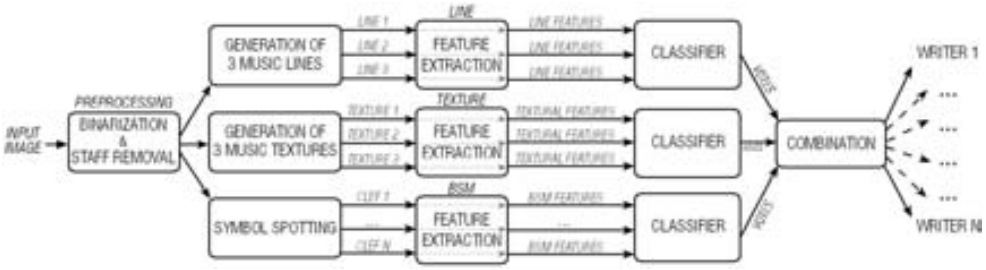
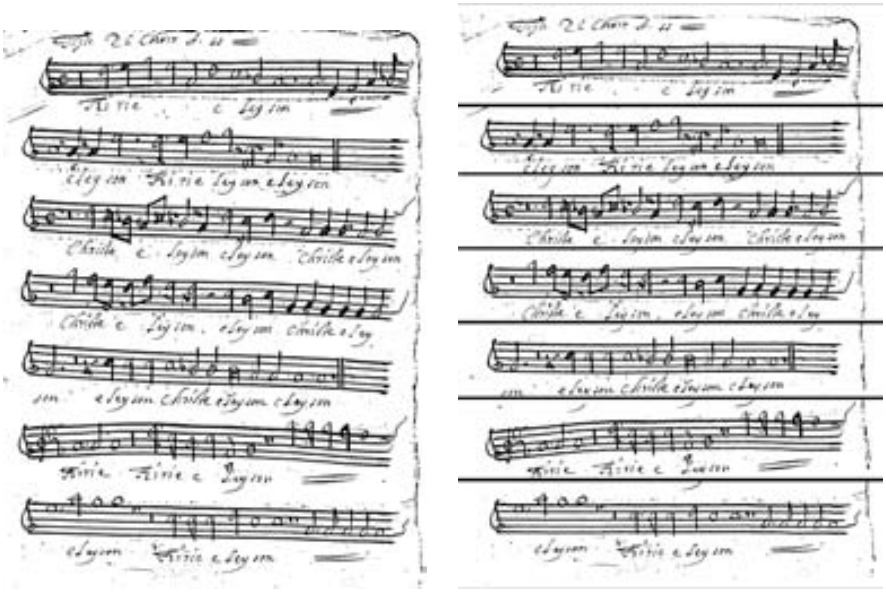


Figure 8.1: Stages of the ensemble architecture for combining the three writer identification approaches.

8.3 Preprocessing

Old music scores have the important restriction of the paper degradation, requiring specific techniques for dealing with noise, the show-through problem and the warping effect. The preprocessing stage consists in the binarization, deskewing, the staff removal and lyrics removal. Well-known methods have been applied for the first two tasks, whereas a new method has been proposed to detect and extract the staff lines, coping with distortions and gaps.

An illustrative example of the whole preprocessing stage applied to a music sheet is shown in Fig. 8.2 and Fig. 8.3. After binarizing the input image (see Fig. 8.2(a)), it is deskewed using the Hough Transform (Fig. 8.2(b)). Notice that each staff has been independently deskewed. Afterwards, the localization and reconstruction of the hypothetical staff lines is performed (Fig. 8.3(a)), and then, contour tracking is used for removing the staff lines. Finally, lyrics are removed (Fig. 8.3(b)). The method is able to detect those pixels belonging to staff lines although there are distortions and oscillations in the staff lines.

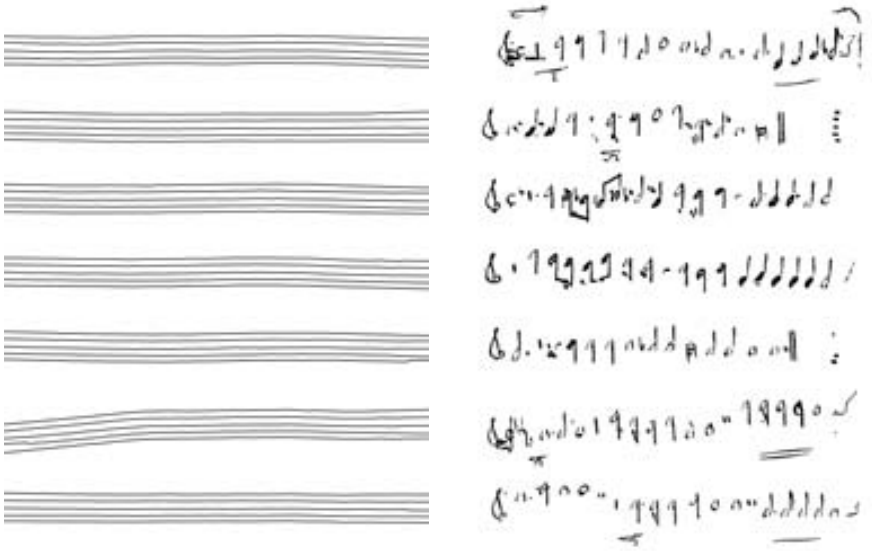


(a) (b)

Figure 8.2: Preprocessing: (a) Original Image; (b) Detected and deskewed staffs

8.3.1 Binarization

First of all, the gray-level scanned image (at a minimum resolution of 300 dpi) must be binarized to separate foreground from background, but with old scores, global binarization techniques do not work because of degradation of the scores. Thus, adaptive binarization techniques are required, such as Niblack binarization method



(a) (b)

Figure 8.3: Preprocessing: (a) Reconstruction of the staff lines; (b) Image without staffs nor lyrics

[Nib86]: the threshold used to classify pixels in black or white is set for each image pixel depending on the neighboring pixels, based on the local mean and local standard deviation of the neighborhood of every pixel (the size of the neighborhood rectangle has been experimentally set to 21x21 pixels). Afterwards, filtering and morphological operations are used to reduce noise.

8.3.2 Deskewing

The second step consists in deskewing the image, so it is rotated so that staff lines are horizontally aligned. The Hough Transform method [BW97] is used to detect lines, and whether this technique is applied to the image, several dots (corresponding to staff lines) will show the orientation of the staff, and consequently, the orientation of the music sheet. If the orientation of these lines is different from 90 degrees (which corresponds to horizontal lines in the Hough Transform space, see equation 8.1), the rotation angle is calculated and the image is rotated.

$$r = x \cdot \cos\theta + y \cdot \sin\theta \tag{8.1}$$

By rotating the whole image, we can not ensure that all the staffs are horizontal. In some music sheets, each staff is oriented in a different angle. For this reason, after deskewing the whole image, projections are used to find the staff sections. Then, the Hough Transform is applied again in each staff region to rotate it in case it is necessary. As a result, each staff has been independently deskewed.

8.3.3 Staff Removal

Although staff lines play a central role in music notation, they cause distortions in musical symbols (connecting objects that should be isolated), making difficult the recognition process. For that reason, staff removal must be performed in order to isolate musical symbols.

The detection of staff lines is difficult due to distortions in staff (lines often present gaps in between), and contrary to modern scores, staff lines are rarely perfectly horizontal. This is caused by the degradation of old paper, the warping effect and the inherent distortion of handwritten strokes (in case they are written by hand). For these reasons, the following process is performed (see Fig. 8.4): After analyzing the histogram with horizontal projections of the image for detecting the location of the staff lines, a rough approximation of every staff line is performed using skeletons and median filters. Afterwards, a contour tracking algorithm is performed to follow every staff line and remove segments that do not belong to a musical symbol. Let us describe the different steps in the following subsections.

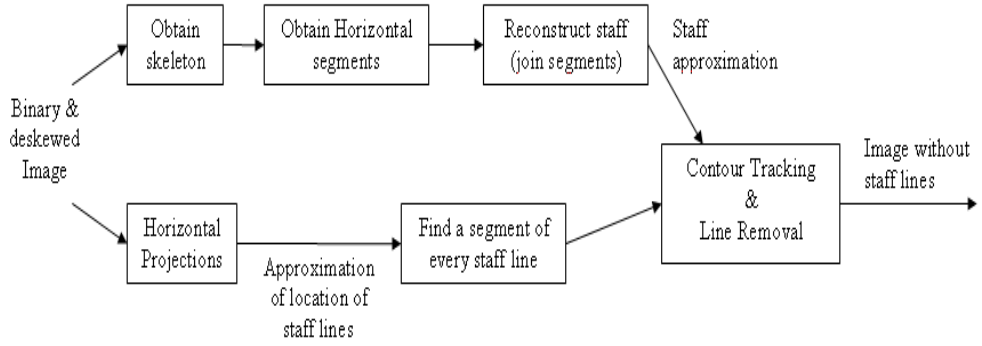
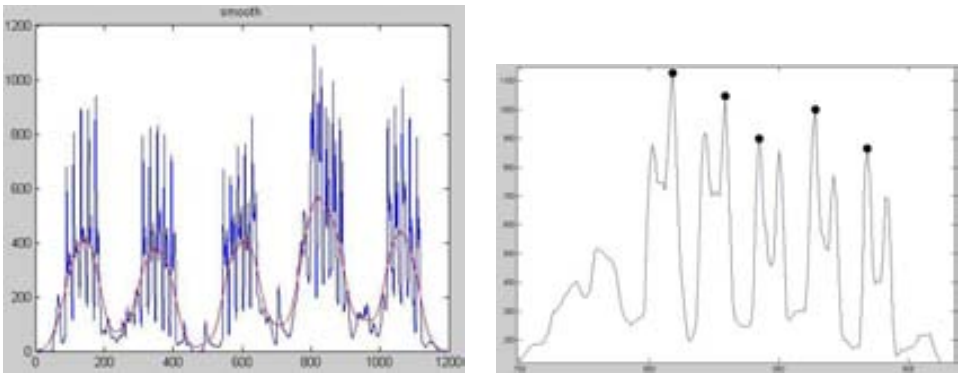


Figure 8.4: Stages of the extraction of staff lines.

Detection of five grouped lines

Since there are deviations in the staff, the detection of staff lines can not be done using horizontal projections (see Fig. 8.5(a)), because sometimes, local maximums do not correspond to staff lines (e.g. two local maximums correspond to one staff line, see Fig. 8.5(b)). Thus, the solution proposed consists in the following steps:

1. Perform a horizontal projection (obtaining an histogram) of the entire score.
2. Smooth the histogram until there is only one oscillation (peak), with only one maximum, for every staff (see the red line in Fig. 8.5(a)).
3. For every oscillation, determine which ones correspond to staffs (a staff has five peaks, corresponding to the five staff lines):
    - If the maximum of a peak is too low, then it is not a staff.



(a) (b)

Figure 8.5: (a) Histogram of the Horizontal Projection of a musical score: the waved-like line corresponds to the smoothing process of the histogram; (b)A segment of the histogram: There are several local maximums corresponding to a staff line, and the dot corresponds to the staff line.

- Smooth the segment of the histogram (SH) corresponding to this staff until there are only five peaks, corresponding to the five staff lines.
- If there are not five peaks, then, it is not a staff.
- If there are five peaks but the distance between them is not constant, then it is not a staff (a staff has five equidistant staff lines).

4. For every staff detected:

   - Obtain five maximums and six minimums in the smoothing histogram (SH) corresponding to this staff.
   - Get the maximum M of the histogram between every two minimums of the smoothing image. Also, this maximum M must be near every peak of the peak. Every maximum M correspond to a staff line (see the dots in Fig. 8.5(b)).

Once we have a rough approximation of the location of every staff line, pixels belonging to every staff line must be determined. The method described next is based on the use of horizontal runs as seeds to detect a real segment of every staff line. Afterwards, a contour tracking process is performed in both directions following the best fit path according to a given direction. In order to avoid deviations (wrong paths) in the contour tracking process, a coarse staff approximation needs to be consulted.

Reconstruction of the hypothetical staff lines

The steps applied to obtain an image with horizontal segments (which will be candidates to form staff lines) are: first, obtain the skeleton of the image, then use a median filter with a horizontal mask, and repeat this process until the last two images are similar.

Thanks to the use of median filters with a horizontal mask, most symbols are deleted from the skeleton of the image, and only staff lines and those horizontally-shaped symbols will remain. Median is less sensitive than mean in front of outliers (extreme values) in the image. Notice that the fact of working with binary images, simplifies the computation of the filter, because the possible values of the pixel are 0 or 1. For that reason, to compute the output value it is only necessary to count the number of 0's (namely $N0$) and 1's (namely $N1$) in the neighborhood and choose the greater value:

$$Output(i,j) = \begin{cases} 0, & if N0 < N1; \\ 1, & otherwise; \end{cases} \tag{8.2}$$

The size of this horizontal mask is constant (experimentally, the best dimensions in pixels are: 1 width $\times$ 9 height), because in the skeletonized image, each line is one pixel-width, so the width of lines in the original image is irrelevant. The process applies median filters deleting iteratively segments that are not horizontal until stability (last two images are similar).

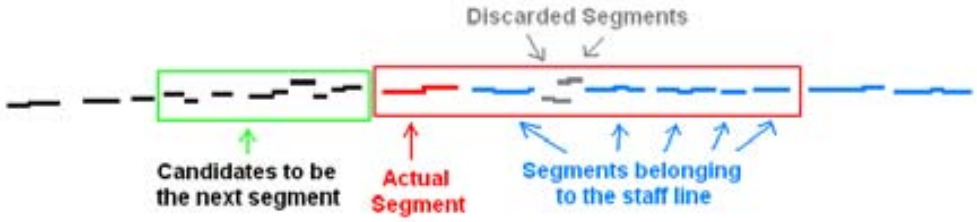


Figure 8.6: Reconstruction of staff lines: Some segments are chosen to be part of the staff line, while other segments are discarded.

Once the image with horizontal segments is obtained, these segments must be used to reconstruct staff lines. For every staff line, the following method is applied for discarding or joining segments, which basically looks the orientation, distance and area of segments:

1. Chose the initial segment of the staff as the larger one.
2. Calculate the slope and the orientation of the initial segment.
3. Reconstruct the staff line, joining segments that are in the left side of the segment. Repeat until the staff line is reconstructed:
   (a) Obtain the statistical mean of the orientations of segments inside a window which contains the segments belonging to the staff line: In order to make

comparisons between orientations of segments, the orientation of the next segment chosen must be compared to the orientation of the last segments which belong to the line (see Fig. 8.6). Thus, the orientation of a single segment will not be so important in order to make comparisons.

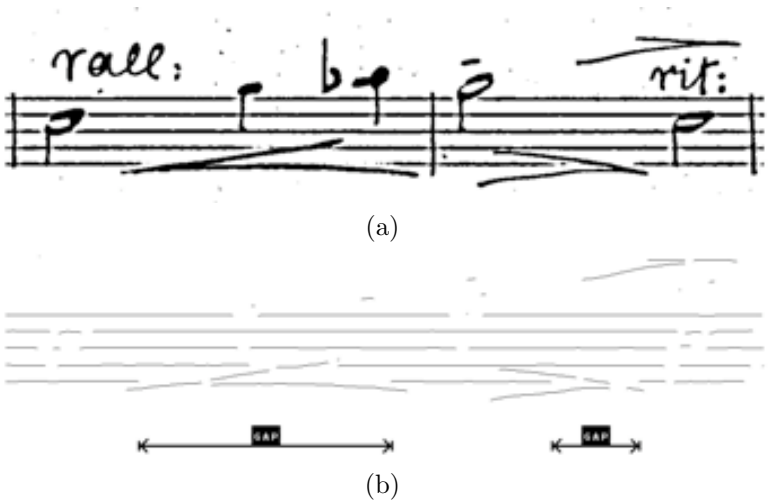
(b) Choose the next segment belonging to the staff line, which is the most *similar* to the actual segment, in terms of orientation, distance, etc. If no segment complies these rules, then return $\emptyset$.

(c) If there is a segment chosen inside the window, then mark those segment as belonging to the staff line and paint the line that joins the actual segment with the chosen one. If there is not a segment inside the window, then, paint the window with a line according to the mean orientation α.

4. Reconstruct the staff line, joining segments that are in the right side of the segment: The method is identical to the one described in step 3.

5. Delete those segments discarded that are near the staff line.

The whole process is described in algorithm 7.



(a)

(b)

Figure 8.7: (a) Original Image (b) Line segments of staff lines with gaps and horizontal symbols.

If there are big gaps in staff lines in presence of horizonal symbols this method could fail and follow a segment of this symbol instead of a segment of the staff line. Figure 8.7(c) shows a big gap with a crescendo marking and Fig. 8.7(d) shows its reconstruction. An initial solution to this problem consists in increasing the size of the slide-window, but it could not work in scores with large deviations in staff lines.

As an example, Fig. 8.8(a) shows the original score suffering from a warping effect and Fig. 8.8(b) shows horizontal segments obtained using skeletons and median filters. The reconstruction of staff lines joining segments is shown in Fig. 8.8(c).

Algorithm 7 Reconstruction of the hypothetical staff lines.

Require: a binary image Z containing only horizontal segments, and the location of the staff lines S
Ensure: binary image with the reconstructed staff lines H

1: **Define** I = morphological opening of Z.
2: **Define** C = connected components of I.
3: **for** each staff line $s \in S$ **do**
4: **Define** R = the larger connected component C that belongs to the staff line s.
5: Calculate the slope and orientation of the segment A as follows:

$$y = m \cdot x + n, \alpha = \arctan(m);$$

6: Reconstruct the staff line, joining segments that are in the left side of the segment.
7: **while** there are candidate segments at the left side of the segment R **do**
8: **Define** α as the statistical mean of the orientations of the n segments inside a window which contains the segments belonging to the staff line, as follows:

$$\overline{\alpha} = [\frac{\sum_{i=1}^{n} \cos(2 \cdot \alpha_i)}{n}, \frac{\sum_{i=1}^{n} \sin(2 \cdot \alpha_i)}{n}];$$

9: Choose the next segment belonging to the staff line:
10: **for** each candidate segment C in the search window **do**
11: Calculate the area, distance to the actual segment A, position of their extremes, orientation of the candidate segment C, and orientation of the line J that joins C with A.
12: Calculate the distance d between orientations of candidates α_i and the actual orientation α of the segment A, as follows:

$$d = min\{abs(\alpha - \alpha_i), 180 - abs(\alpha - \alpha_i)\}$$

13: **end for**
14: Return a segment R that complies:

$$\begin{cases} distance(R \rightarrow A) < threshold; & and \\ area(R) > threshold; & and \\ orientation(R) \simeq orientation(A); & and \\ orientation(J) \simeq orientation(A); \end{cases}$$

If no segment complies these rules, then return $\emptyset$.
15: If there is a segment chosen inside the window, then mark those segment as belonging to the staff line and paint the line that joins the actual segment with the chosen one. Otherwise, paint the window with a line according to the mean orientation α.
16: **end while**
17: Reconstruct the staff line, joining segments that are in the right side of the segment: The method is identical to the one described in steps 7 to 16.
18: Delete those segments discarded that are near the staff line.
19: **end for**

Contour Tracking

After the obtention of the reconstructed staff lines, the contour tracking process can be performed following the best fit path according to a given direction. The aim is to remove pixels belonging to staff once their location is roughly determined. The main idea of the contour tracking is to select the longer segment of every staff line, and then perform contour tracking of the staff line. The tracking is performed in both directions, following the contour of the line, and consulting the image with the hypothetical staff lines whenever is required (e.g. in presence of gaps, or bifurcations). The process is described nest.

For every staff line:

1. Take a window that includes the staff line and obtain the width of this staff line: perform a Run Length Smearing vertical and catch the longer segment. The width of that staff line W will be the statistical mode of the width of this segment.

2. Perform a vertical Run Length Smearing with a segment of length = W, and detect a segment SG longer and closer to the horizontal line detected in the histogram of horizontal projections.

3. Take the segment SG and perform the contour tracking towards the left direction. Repeat until the beginning of the image:

   (a) Take a little column in the left side of the segment, and detect positions of the pixels which belong to the contour in that column:

   - If there is no pixel in the little column but there are pixels in a section near it, then determine if there is a change of line or not (depending on the distance and orientation).
   - Chose the connected component in the column with bigger area and closer to the actual positions of the segment. Then, calculate its extremes. If those positions are too far from the positions of the actual segment, or they are too far from the hypothetical reconstructed staff line, then reject those component.

   (b) If points are returned, mark them as belonging to the staff line. If no points are returned, then mark next points, depending on the hypothetical reconstructed staff line.

4. Take the segment SG and perform the contour tracking towards the right direction until the end of the image: The method is identical to the one described in (c).

Notice that whether there is no presence of staff line (a gap), the contour tracking process is able to continue according to the location of the reconstructed staff line.

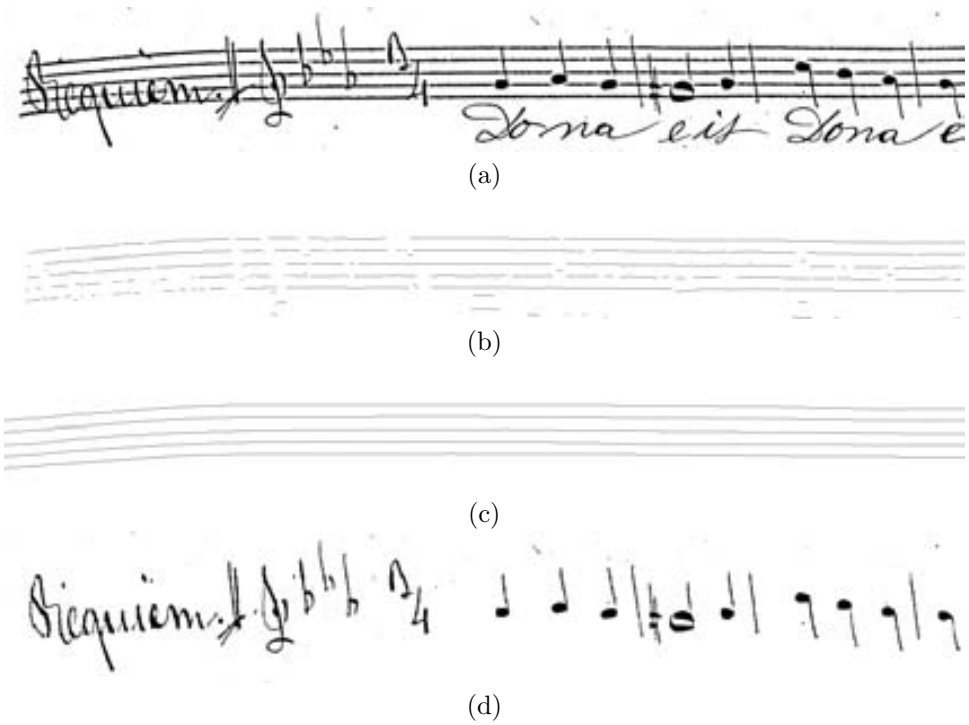


Figure 8.8: (a) Original Image; (b)Horizontal segments of the score; (c) Reconstruction of the hypothetical staff lines. (d) Image without staff lines nor lyrics.

Staff Removal

Concerning line removal, we must decide which line segments can be deleted from the image, because whether we delete staff lines in a carelessly way, most symbols will become broken. For that reason, only those segments of lines whose width is under a certain threshold (experimentally set to 1'2 * *width of staff lines*) will be removed. As it has been commented, width of staff lines has been calculated in the contour tracking process, using the statistical mode of line-segments.

Figure 8.9 shows some examples of line removal: Figure 8.9(a) is the original image, where in Fig. 8.9(b) we can see how in presence of a gap, the process can detect next segment of staff line to continue; in Fig. 8.9(c) a symbol crossing the line will keep unbroken, because the width of the segment is over the threshold.

In this level of recognition, it is almost impossible to avoid the deletion of segments of symbols that overwrite part of a staff line (they are tangent to staff line, see Fig. 8.9(d)) and whose width is under this threshold, because context information is not available. Fig. 8.8(d) shows an example of the results of the staff removal module, which can cope with deviations in staff lines.

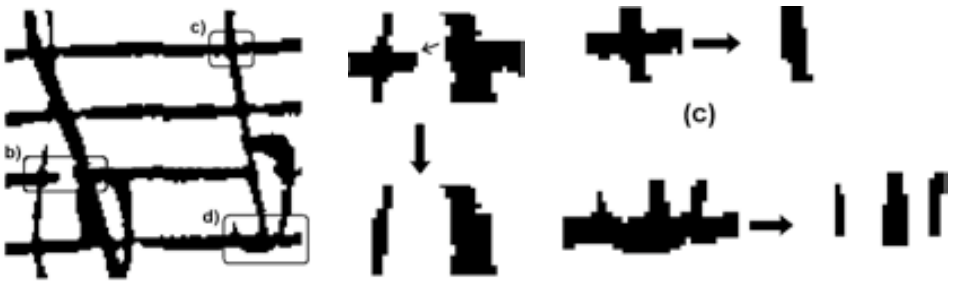


Figure 8.9: Examples of Line Removal in Contour Tracking process. a) Original Image, b) Gap in line, c) Symbol crosses the staff line, d) Symbol is tangent to staff line: Symbol becomes broken.

8.3.4 Text removal

As we have discussed in Chapter 1, text and lyrics are not taken into account for our writer identification system. For this reason, they must be removed from the image. Although text-symbol separation can be an extremely difficult problem (e.g. when text and symbols are touching and overlapping), and thus, an intensive research should be done, it is out of the scope of this work. We have used the following hypothesis: each connected component which is not touching a staff line, will be labeled as lyrics and removed from the image. Notice that this hypothesis is valid in most cases, but it is not always true. For this reason, the resulting image must be supervised in order to correct any music symbols wrongly removed, and also, removing any text that is touching the staff. An example can be seen in Fig 8.8(d), in which the word *Requiem* must be manually removed.

8.4 Combination of Classifiers

Once the staffs and lyrics are removed from the image, the different writer identification approaches perform a classification of the input image.They first perform the specific preprocessing (normalization, texture generation, clef detection), and then the extraction of features is performed. The first approach will apply the symbol detection technique (a combination of the BSM and the DTW-based methods) to detect clefs and extract the BSM features; the second one will generate three music lines and will apply the 98 local features; and the third one will generate three music textures and will apply the 92 textural features.

Combination of the results obtained for a single classifier. Due to the fact that from every music page we obtain three music line images, three texture images or n music clefs, all these elements belong to the same music sheet, and consequently, they should be only assigned to one class. This might be performed combining the classification results of the n elements using the Majority Voting and the Borda count method. In both methods, each line is classified to the k candidate classes (using the k-NN classifier). The list with the three candidates is sorted so that the first candidate has obtained the higher confidence rate. The difference between Majority Voting and Borda Count is that for the Majority voting, each candidate adds one vote to the corresponding class. Contrary, in the Borda Count method, the first ranked candidate adds more votes to the class than the last ranked candidate, which adds the lower number of votes.

A comparative example can be seen in Fig.8.10. The three input lines (belonging to the same sheet) are classified. The first line has been classified as classes A, C and B, the second one has been classified as classes D, B and C, and the third one as classes E, A and B. In the Majority Voting method, each candidate gives exactly one vote to each class, so the three lines are classified as class B because class B has the maximum total amount of 3 votes. Contrary, in Borda Count the first candidate gives three votes, the second candidate gives two votes, and the third candidate gives only one vote, so the three lines are classified as class A because it has a total amount of 5 votes. Notice that the three lines should be classified as different classes depending on the combination method used. It must be said that in this example, if no combination was used, the first line should be classified as class A, the second one as class D and the third one as class E.

Combination of the results obtained for the three classifiers. In our experiments, each writer identification approach performs the voting step, using the Euclidean Distance, with the k-NN classifier [RPD01], and the Majority Voting or Borda Count method.

The combination is performed as follows. Firstly, for the line-features approach, the three input lines will give votes to the k nearest neighbor classes. Secondly, the three input textures will also give votes to the k-NN classes. Thirdly, using the same procedure, every accepted clef will also perform the voting. Finally, all the votes are counted, and the input music sheet will be classified as the class which has received the major number of votes.

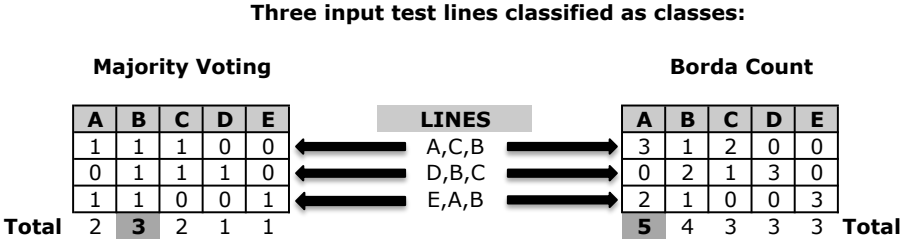


Figure 8.10: Majority Voting and Borda Count combination example. The three input lines are classified as a ranking of candidates. Each candidate gives 3, 2 or 1 votes if Borda Count is applied, whereas they always give exactly one vote when using Majority Voting. One can see that the three input lines will be classified as class B using Majority Voting or A using Borda Count.

Notice that, contrary to the first two approaches (in which just 3 lines or 3 textures can vote), the number of votes from the third approach depends on the number of detected and accepted clefs. In most of the cases, there are more than three clefs per page, and consequently, the symbol-dependent method has a greater influence in the final identification than the line and textural approaches. This greater influence is obviously desired because the symbol classifier has a more accurate identification rate than the other two approaches.

8.5 Experimental Results

In this section, we analyze the results obtained applying the staff removal approach. Then, we compare and discuss the results obtained for each writer identification approach. Finally, we analyze the classification results obtained for the different combinations of approaches. These results are discussed next.

8.5.1 Staff Removal

We have tested the proposed method with 200 images of scores scanned from the archive of the Seminar of Barcelona and the Archive of Canet de Mar (this database is fully described in the Appendix). After the binarization and deskewing of the image, the staff lines are located and the hypothetical staff lines are reconstructed. Afterwards, the contour tracking process is used for removing the staff lines from the music score. In Table 8.1 we can see the results of the application of this method to the database, showing that 76.8% of the staff lines are completely removed, and 10.8% are almost completely removed. Contrary, 12.3% of the staff lines are partially or not removed, requiring a manual removal.

These results show that the proposed method has several limitations. Firstly, it must be noticed that in some cases, the author has written lyrics on a staff, and them some text strokes cause too much distortion in the staff lines. In these cases, the staff detection module can fail in the search of five maximums in the histogram and could not be able to detect a staff with text (see Fig. 8.11). Secondly, and concerning staff

Staff Removal of 200 pages	Perfectly Removed	Almost Removed	Partially Removed	Not Removed
1198 staffs, 5990 staff lines	4602	650	180	558
TOTAL	76.8%	10.8%	3%	9.3%

Table 8.1: Staff removal results of 200 pages: The number and rate of the staff lines which have been perfectly, partially and not removed are shown.

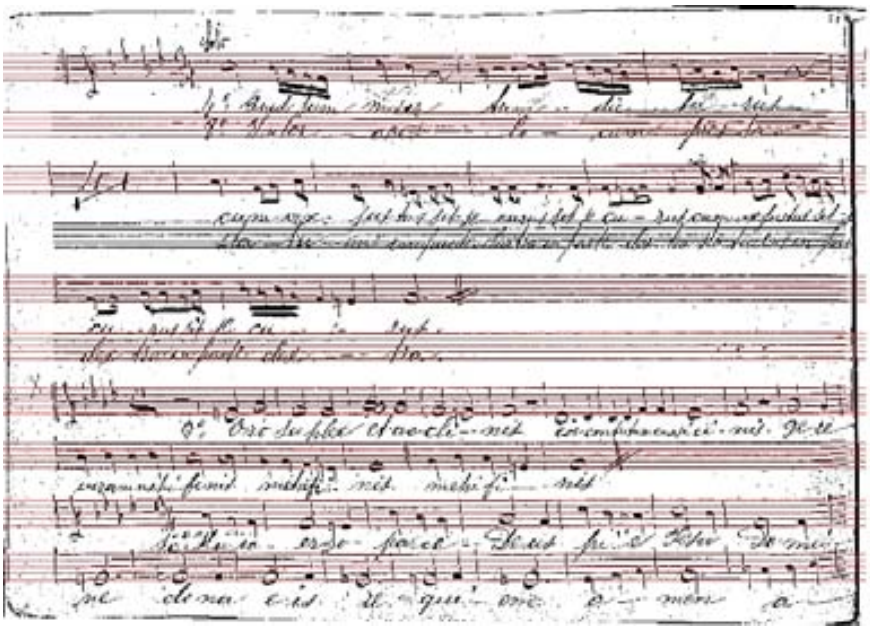
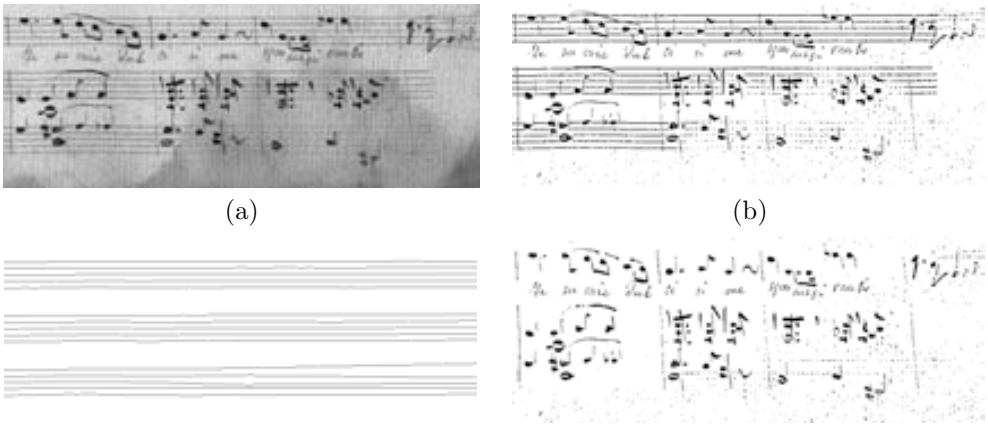


Figure 8.11: Detected Staff lines: There is one staff missing

lines reconstruction, although most staff lines can be well reconstructed, in some cases, a horizontal symbol is drawn over a staff line and causes the staff reconstruction to follow wrongly this symbol. As a consequence, the staff removal method only removes a section of the staff line. Finally, when applying the method to very degraded music sheets, the staff line can not be completely removed (see Fig. 8.12). Some examples are the music scores with show-through (the staff lines of the backpage are disturbing the detection and reconstruction of the staffs belonging to the actual processed sheet), those music scores whose staff curvatures are constantly changing (the staff can not be correctly deskewed, and consequently detected), or the music scores with some missing sections in the staff lines. In such these cases, the reconstruction of the hypothetic staff lines can not be correctly performed, and consequently, the staff is not completely removed.



(a) (b)

(c) (d)

Figure 8.12: Staff reconstruction: (a) Original Image; (b) Binarized image; (c) Staff reconstruction: The final part is not correctly reconstructed; (d) Staff Removal: The end of section is not completely removed

8.5.2 Comparison of the Three Proposed Approaches for Writer Identification

Before combining the three writer identification approaches proposed in this thesis, they will be individually compared. Firstly, we will compare the two symbol-independent methods. Afterwards, we will compare the symbol-dependent method versus the two symbol-independent ones.

Comparison of Line Features versus Textural Features

We have compared the two *symbol-independent* writer identification methods. The first is based on the extraction of 98 typical features for handwritten text recognition. The second one is based on the extraction of textural features from texture images. Table 8.2 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 features), whereas, the line features get higher results for a database with more 15 or 20 writers (76% versus 73% of w.i.r. with 20 writers for line features and textural features respectively). Due to the fact that a writer identification rate of 76% is not a significant improvement of the 73% rate, we can affirm that textural features reach similar identification rates for this specific database. For this reason, the final decision should be made after testing the two methods on a big database or writers.

Number of Writers	W.I.Rate 98 Line Features	W.I.Rate 92 Textural Features
5	84% (14.66)	96% (7.84)
10	82% (13)	86%(11.76)
15	78% (8.66)	75% (10.45)
20	76% (8.43)	73% (11)

Table 8.2: Classification Results: Writer identification rates using 98 line features and 92 textural features for different database sizes. The score is computed by means of stratified five-fold cross-validation, testing for the 95% of the condence interval with a two-tailed t-test

Comparison of *symbol-independent* approaches versus the *symbol-dependent* approach

As it has been discussed, the comparison of the results of the two *symbol-independent* approaches shows that both approaches reach similar performance (73% and 76% of writer identification rate) for this specific database. Contrary, the results obtained by the *symbol-independent* approach (92.5%) demonstrate that the performance is significantly increased. It must be noticed, that the 92.5% of identification rate is obtained using only 16 writers, and should not be compared to the identification results of the two methods for 20 writers. In any way, the results obtained by the *symbol-dependent* approach outperforms the other approaches, even when they are applied to 10 writers. Unfortunately, the symbol-dependent method can not be always applied.

As a summary, we can say that although the *symbol-dependent* approach based on symbol recognition methods obtains very high identification rates, it can not be used for all the writers, and for this reason, a combination of the results of these three approaches should be the optimal choice.

8.5.3 Final Writer Identification Results

The ensemble architecture proposed in this Chapter 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 three individual writer identification methods. First of all, the input image is preprocessed. Then, three music lines and three textures are generated from the image without staff lines, and the symbol spotting technique is used for detecting the music clefs. Afterwards, the corresponding features are computed (line, textural or BSM features). For the classification, the three music lines give votes to the nearest neighbor classes using the Borda Count method, the Euclidean distance, and the 5-NN classifier (which has shown to obtain the best results). Similarly, the three textures and the detected clefs give votes to the neighbor classes. Finally, the votes obtained for each approach are summed, and the maximum value will indicate the final labeled class. The Borda Count method has been chosen, because it has usually shown better results than the Majority Voting method in the individual experiments of the writer identification approaches.

Experiment	#	Test 1	Test 2	Test 3	Test 4	Test 5	Total (Average)
Music Lines (98 line feat.)	20	65%	75%	95%	70%	70%	**75% (10.3)**
Resize Textures (92 textural feat.)	20	65%	55%	80%	85%	80%	**73% (11)**
Symbols (Clefs) (25x25 BSM feat.)	**16**	93.7%	87.5%	100%	87.5%	93.5%	**92.5% (4.6)**
Lines & Textures	20	80%	95%	100%	95%	90%	**92% (6.6)**
Lines & Clefs	20	75%	85%	95%	90%	85%	**86% (6.5)**
Textures & Clefs	20	80%	90%	95%	100%	90%	**91%(6.5)**
Music Lines & Textures & Clefs	20	85%	95%	100%	100%	95%	**95% (5.4)**

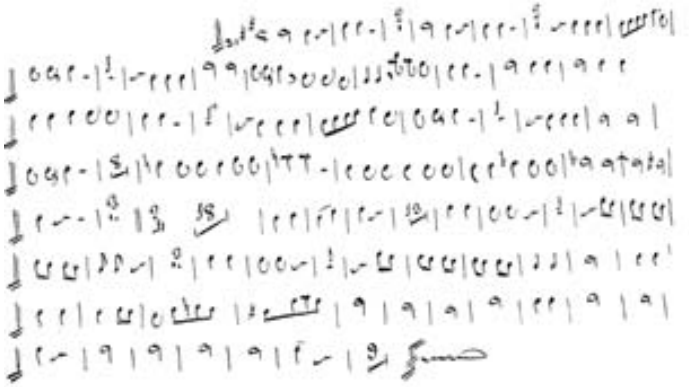
Table 8.3: Combination of Results of the three writer identification approaches (Number of writers = #). We use 5-fold cross-validation, a 95% of condence interval, and the 5-Nearest Neighbor classifier.

For the experiments, we have used 5 test subsets, randomly chosen, containing one page per writer. For a fair comparison, this sets are the same than have been used for testing the writer identification methods based on music lines and textures. It must be noticed that for this experiment, the symbol-dependent writer identification method has not classified four writers of the database, whereas the other two methods have results for the whole set of 20 writers. Thus, the symbol-dependent approach adds no vote to the classes, and the final identification is performed with only the votes of the other two approaches (lines and textures-based ones).

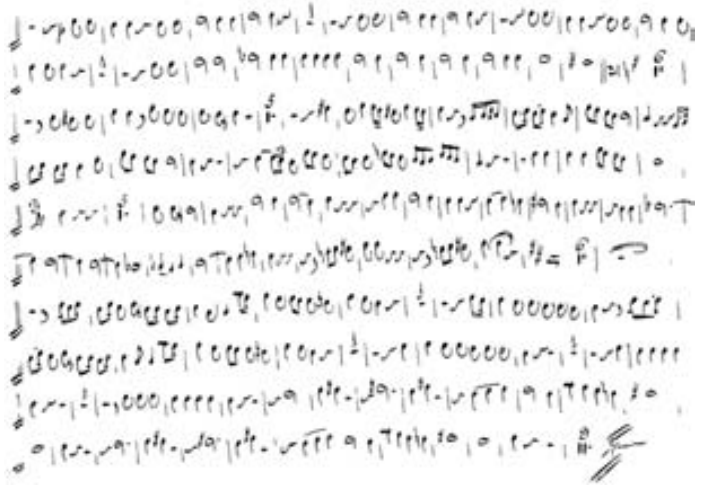
Table 8.3 shows the writer identification rate (w.i.r.) for each one of the 5 tests, and the final identification rate. The first two rows show the identification rates when using line features (75%), textural features (73%). The third row shows the w.i.r. obtained using BSM features for the detected clefs (92.5%), but this value can not be compared with the previous ones, because the size of the database is not the same. Next rows show the results of different combinations of approaches. The combination of line and textural features reaches similar results (92%) than the combination of textural features and BSM features from clefs (91%), whereas the combination of line and BSM features from clefs has lower improvement (86%) when comparing with the individual classifications.

It must be noticed that the proposed architecture extracts information about the music symbols, avoiding the dependence of 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 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 approach, texture images are generated by randomly selecting symbols, and then, resizing them. In the third approach, music clefs are the symbols used for extracting features. Thus, we minimize the effect of the different number and kind of symbols in the music sheets. As an example, Fig.8.13 shows two music scores written by the same writer, which have been correctly classified

as belonging to the same class. One can see that although the density and the kind of symbols is different, the writer style is very similar (compare the shape of notes, clefs, ending signature).



(a)



(b)

Figure 8.13: 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.

It is remarkable the high performance of the combination of the three approaches (the last row of the table), reaching a 95% of writer identification rate. It is the best w.i.r. of all, higher than the rates obtained from the three individual approaches. Thanks to the combination of results, when one approach has misclassified a test image, the other two approaches can compensate this misclassification. In fact, it might occur that even though the three classifiers had wrongly classified an input test image, the image could be correctly classified thanks to the combination of results. For example, for an input image X belonging to the class B, take that the first classifier gives the following votes $[A = 10, B = 8, C = 2]$, the second classifier votes

$[C = 8, B = 7, A = 5]$, and the third one votes $[C = 9, B = 8, A = 3]$. Notice that the three classifiers have misclassified the test image X (the first approach decides that A is the first ranked class, and the second and the third ones classify it as class C), but they three agree that the second ranked class is B. After summing all the votes, we obtain $[B = 23, C = 19, A = 18]$, and consequently, the input image X has been correctly classified as belonging to class B.

As a summary, we can affirm that for this specific database, the combination of the three approaches is the most suitable choice.

8.6 Conclusions

In this Chapter we have presented the preprocessing step, which is common in the three writer identification approaches proposed in this work. It consists in binarizing, deskewing and removing the staff lines and lyrics. Results show that it is a good solution for old handwritten documents, although it has some limitations when working with very degraded music sheets. Obviously, a more acurate binarization method should facilitate the staff removal stage.

Afterwards, we have described the ensemble method used for combining the three writer identification approaches (text lines, textures, and music clefs). The first approach takes the preprocessed image, detects the music clefs and extracts the BSM features. The second approach generates the three music lines and extracts 98 line features, and the third approach generates the three texture images and computes 92 textural features. The next step consists in the ensemble of these three approaches, which has been performed using the Borda Count combination method. Thus, each classifier gives votes to the nearest neighbor classes according to the confidence rate (e.g. the first nearest neighbor obtains more votes than the second nearest neighbor). Finally, the input music sheet will be classified as the writer which has received more number of votes.

Concerning the comparison of the individual approaches, the results show that the results obtained by the two symbol-independent approaches are quite similar (73% and 76% of writer identification rate), whereas the symbol-dependent approach obtains the highest identification rates.

The final results demonstrate that the combination of the three approaches outperform the individual approaches (95% of writer identification rate), demonstrating the suitability of the proposed ensemble architecture, being the optimal choice.

Chapter 9

Conclusions and Future Work

In this Chapter, we summarize the contributions of this thesis to the field of writer identification applied to music scores. Afterwards, we discuss the performance of the methods proposed and their limitations. Finally, future work is presented.

This thesis has addressed the task of writer identification of music scores, as an example of graphic documents. As a result, we have proposed three different approaches and the ensemble architecture for combining them. The ensemble architecture has demonstrated to be the best choice, obtaining very high identification rates. As far as we know, this is a pioneer work addressing the problem of writer identification in handwritten documents of graphical languages. We believe that we have done a step forward in the field of graphics recognition.

This last Chapter is organized as follows. In Section 1, the summary and contributions of this work are described, whereas in Section 2, we discuss about the advantages and limitations of the proposed methods. Finally, Section 3 proposes future work.

9.1 Summary and Contribution

The main contribution of this thesis has been the proposal of a writer identification architecture for old handwritten music scores. It consists in a ensemble architecture that, after the preprocessing of the image (in which it is binarized, desked and staffs are removed), it combines the results of the three different writer identification approaches (a *symbol-dependent* method and two *symbol-independent* methods) for the final classification. Let us summarize the main contributions.

Staff Removal The three writer identification approaches require a common preprocessing step, consisting in the binarization, deskewing and staff removal. In this stage, a novel method for detecting and removing the staff lines in old handwritten music scores has been proposed, able to cope with gaps, deformations due to paper degradation and the warping effect. We have proposed a line tracking approach

combined with projection profiles. The method is able to detect staff lines although they follow a curvilinear path, and also it is able to reconstruct objects after lines are removed.

Writer identification based on symbol recognition The first approach consists in a writer identification method based on symbol recognition, which detects the music clefs, and extracts features about their shape. It is a non-supervised approach, which takes use of several novel symbol recognition methods and a symbol-detection technique.

Referring the symbol recognition methods, they have been specially designed for hand-drawn symbols, obtaining robust approaches in front of the typical distortions of handwritten graphical documents. The first one is based on the Dynamic Time Warping algorithm, which has been adapted to bidimensional data. The proposed method computes a column sequence of feature vectors for each orientation of the two symbols and computes the DTW distance taking into account the perpendicular alignment. In the second one we have proposed the Blurred Shape Model (BSM) descriptor, which encodes the probability of pixel densities of the image regions. In addition, an evolution of the BSM has been presented, which uses a correlogram structure for obtaining a rotation invariant descriptor.

Concerning the detection of clefs, we have proposed a symbol-detection technique, which uses the combination of the BSM descriptor and the DTW-based method. Finally, the BSM features computed from the music clefs have been used for identifying the writer of the music sheet.

Writer identification based on line features The second approach consists in a writer identification method based on music lines. It is an adaptation of the text-independent writer identification method for text documents defined by Hertel and Bunke [HB03]. The main contribution has been the specific preprocessing and normalization of the music scores, and the adaptation of the features to graphic documents (such as music scores). Afterwards, line features have been extracted (consisting in basic measures, connected components, contours and fractal features), which are consequently used for identifying the writer.

Writer identification based on textural features The third approach consists in a writer identification method based on textural information. We have adapted the approach defined by Said et al. [STB00] for text documents to music scores. The main contribution has been the proposal of several approaches for the generation of music texture images. Every approach uses a different spatial variation when combining the music symbols to generate the textures. After the computation of textural features (consisting in Gabor filters and Grey-scale Co-ocurrence Matrices), the identification of the writer has been performed.

Validation Framework Finally, the last contribution has been the construction of a validation framework for writer identification. A database of old music scores has been obtained by the digitalization of old handwritten scores (from the 17th to 19th

centuries) from the archive of the Seminar of Barcelona, Terrassa and Canet de Mar. Although the database has been performed for the task of writer identification, it can also be used for other tasks, such as binarization, staff removal, symbol recognition and optical music recognition.

9.2 Discussions

Musicologists identify the writer (or composer) through a deep analysis of the music score. They perform a recognition and interpretation of the whole music score, taking into account all the discriminant properties in handwriting music notation which have been described in the Introduction. In addition, they analyze the rhythm, melody, and harmony of the composition for obtaining information about the music composition style. This kind of procedure is classified as symbol-dependent writer identification method, because they must recognize the different elements in the score in order to identify the author.

Similarly, an automatic symbol-dependent writer identification approach should ideally recognize and understand the semantic information of the whole music score. 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 these reasons, we have proposed a symbol-dependent writer identification method that uses a small part of the methodology used for musicologists. Moreover, we have proposed two symbol-independent methods, modeling the global characteristics of the image, avoiding the dependence of a good recognizer.

Referring the preprocessing step, it must be said that although experimental results show that the staff removal algorithm has good performance, it has some limitations when it is applied to very degraded documents. Obviously, the reconstruction of the hypothetical staff lines has some problems when there are too many or two few segments to join. In addition, the staff removal approach strongly depends on the performance of the binarization technique, because a poor binarization might generate too much noise, making more difficult the removal of the staff lines. Probably, better results could be obtained using more accurate binarization techniques.

Concerning the three proposed symbol recognition methods, they have shown very high recognition rates on different hand-drawn data sets, outperforming the state of the art approaches. The DTW-based method is the best choice for the recognition of symbols with a high variability, but it is more sensible to noise than the BSM descriptors, and requires a good segmentation of the symbol. Contrary, the BSM descriptors have lower performance when classifying symbols with an important intra-class variability, but they can be applied to several kind of problems (e.g. symbols with noise and gaps). In addition, they are very fast to compute, being suitable for symbol detection problems.

The two symbol-independent approaches (music lines and Resize textures) for writer identification have shown to obtain similar performance when applied to music scores. Some feature selection methods have also been applied in order to improve the performance. Results show that although they reach similar identification rates,

the dimensionality reduction is very significant, requiring less than a third part of the set of features.

The symbol-independent approaches are very robust, because a recognizer is not needed. Contrary, the symbol-dependent approach requires an accurate symbol-detection and recognition method, otherwise, clefs are not correctly detected nor segmented. In fact, experimental results show that there is an important amount of false positives and false negatives. Consequently, although the method has the highest discriminatory power of the three approaches, the performance of the method decreases with a poor recognition step.

Finally, it must be concluded that the experimental framework proposed combines the results of the three approaches, obtaining better results than the individual ones, and demonstrating the suitability of the proposed ensemble architecture for writer identification. It must be noticed that the experimental results show the performance of the methods when applied to this specific data set of 200 music scores, and consequently, the results could potentially be different for other datasets.

9.3 Future Work

Despite the advances performed in the identification of the writer of music scores, the limitations discussed in the previous Section show that there are still open issues to be further addressed.

Text and Lyrics

In order to improve the detection of text and lyrics, several possibilities could be tested:

- Analyzing the size of the bounding box of the connected components: The size of bounding box for text is different from the size of musical symbols.

- Orientations of strokes in text are changing constantly, so the Structural Tensor [GL96] could be used to find sections with a lot of changes in orientation in their strokes.

- Fractal Dimension: As it has been effectively applied for writer identification approach based on music lines, the fractal features could show that the function of text is different from the function of symbols.

- Dictionary: There is a finite set of words corresponding to *dynamics* (e.g. *allegro, forte, ritardando...*), thus, a dictionary could be used to distinguish musical words from lyrics and symbols.

Once the text and lyrics are detected, they can be extracted as used as the input of a writer idenfitivation method for text [SB07a]. Thus, the identification approach for text could be combined with the identification approach based on music notation.

Textural Information

Concerning the writer identification method based on textural information, the work can be extended as follows:

- Other approaches for generating textures could be proposed in order to obtain more discriminative textural images. A good idea could be a combination of the AspectRatio and Resize ideas. It has been shown that the Resize textures reach the highest identification rates. In order to avoid the high frequencies that interfere in the representation space, the resized symbols could be randomly distributed along the texture image.
- Other textural features could be applied to improve the final classification rate. In fact, some works have shown that wavelets obtain better performance than Gabor features [HS08].
- The combination of the textural features of the five different texture images (Basic, TextLine, Random, AspectRatio and Resize) could be performed, with the consequent application of feature selection or combination methods.

Symbol-dependent Methods and OMR

Concerning the writer identification approach based on symbol recognition methods, more features should be added to the approach in order to do it extensible to scores with few music clefs. As it has been discussed in Chapter 5, information about music notes, rests, ending signatures and accidentals could be combined and added to the information extracted from clefs. For this purpose, instead of a symbol detection method used for detecting all these music symbols, a full optical music recognition system should be developed. In this sense, the following options could be tested:

- Grammars have been defined for helping in the recognition, discarding false detections and helping with ambiguities [CR95]. In this sense, we have proposed a grammar for the OMR task (see the Appendix B).
- Hidden Markov Models (HMM) have also been used for the task of OMR [Pug06]. HMM are robust statistical models to model sequence observations, widely used in handwriting recognition. Therefore, they are able to describe other types of languaguages such as the musical one.

Symbol Recognition methods

Referring the proposed BSM and Dynamic Time Warping-based symbol recognition methods, further work could be focused on:

- The development of DTW variations for decreasing the time complexity of the algorithm, being inspired in the DTW proposals for massive datasets [KP99].
- The addition of more features to the column vector could reduce the sensibility to noise. A good option could be the combination of the features of the BSM descriptor with the features of the column vector computed for the DTW method. As a result, the advantages of the two methods could be obtained.

Staff Removal

The staff removal step could be improved by performing a better reconstruction of the hypothetical staff lines. As it has been commented, the contour tracking process will success wherever the hypothetical staff lines are perfectly reconstructed. Due to the fact that the reconstruction of horizontal segments into staff lines can fail if there are big gaps or distortions in the staff, this method could be improved as follows:

- Looking the five parallel staff lines at the same time when reconstructing. Thus, if there are deviations or ambiguities, the system can look the path that has been followed by the other four lines, and then choose a path trying to keep five lines equidistant. This solution will improve the reconstruction module when working with scores with distortions and warping effect. Contrary, this constraint must be relaxed with staff lines written by hand, because sometimes, five lines are not equidistant enough.

- Constructing a graph with horizontal segments as nodes. Then, every reconstructed staff line will be the output of an algorithm which follows the best fit path (with backtracking). An example of contour tracking as a best fit path is described in [BB82].

Classification

The ensemble architecture that combines the different writer identification approaches uses the Euclidean distance, the k-NN classifier and the Borda Count combination method. In this sense, the work could be extended by using more sophisticated classification methods. An interesting option is the combination of the features extracted by the three approaches in a single set of features, and then, the classification can be performed using Support Vector Machines, Principal Component Analysis or Adaboost [Kun04].

Framework

Finally, the database used for the experiments should be increased. As it is commented in the Appendix, there are more than 500 music pages obtained from more than 50 composers. Unluckily, the compositions of the same composer are not written by the same writer (scribes and copyists could do it). For this reason, a bigger database should be obtained in order to perform further experimental results concerning writer identification.

Appendix A

Databases

Due to the lack of public available databases of old handwritten music scores, several databases have been created for validating the proposed methodology. The first data set is composed of old handwritten music scores, for validating the proposed architecture for writer identification. The second one is composed of hand-drawn music clefs and accidentals, and has been used for validating the symbol recognition methods. The two databases are described next.

A.1 Old Handwritten Music Scores

The data set of old music scores is extracted from a collection of music scores of the 17th, 18th and 19th centuries. They have been obtained from three archives in Catalonia (Spain): the archive of Seminar of Barcelona, the archive of Terrassa, and the archive of Canet de Mar. The data obtained from the archive of Seminar of Barcelona is composed of 19 music sheets drawn by 3 different writers, whereas 102 music sheets from 6 different writers are obtained from the Seminar of Terrassa. From the archive if Canet de Mar, 560 music sheets have been scanned from 50 different composers. In total, we have obtained a database of 681 pages from 59 different composers.

The music sheets have been scanned using a flatbed scanner, and stored in bitmap format. They have been captured in gray-scale at a resolution of 300 dpi, which is enough for capturing the information contained in the image.

Subset used for the Writer Identification experiments

A small subset of the database of music scores has been used for the writer experiments, because although there are a total of 59 composers, there were many scribes and copyists, and consequently, there are not 59 different writers. For this reason, a detailed analysis of the music sheets has been performed, in order to detect the different handwriting styles. In some cases, a few number of music pages was written by a specific writer, and consequently, they can not be added to the experimental set.

As a result, we have selected 10 pages for each one of 20 writers (19 from the

Seminar of Canet de Mar, and one from the Seminar of Barcelona), obtaining a testing set of 200 music pages. In Figures A.1, A.2, A.3, A.4, A.5, two music sheets of different composers are shown. Notice the important differences in the handwriting style between different writers (e.g. curvature of the writing, shape of clefs, notes, rests, etc.).

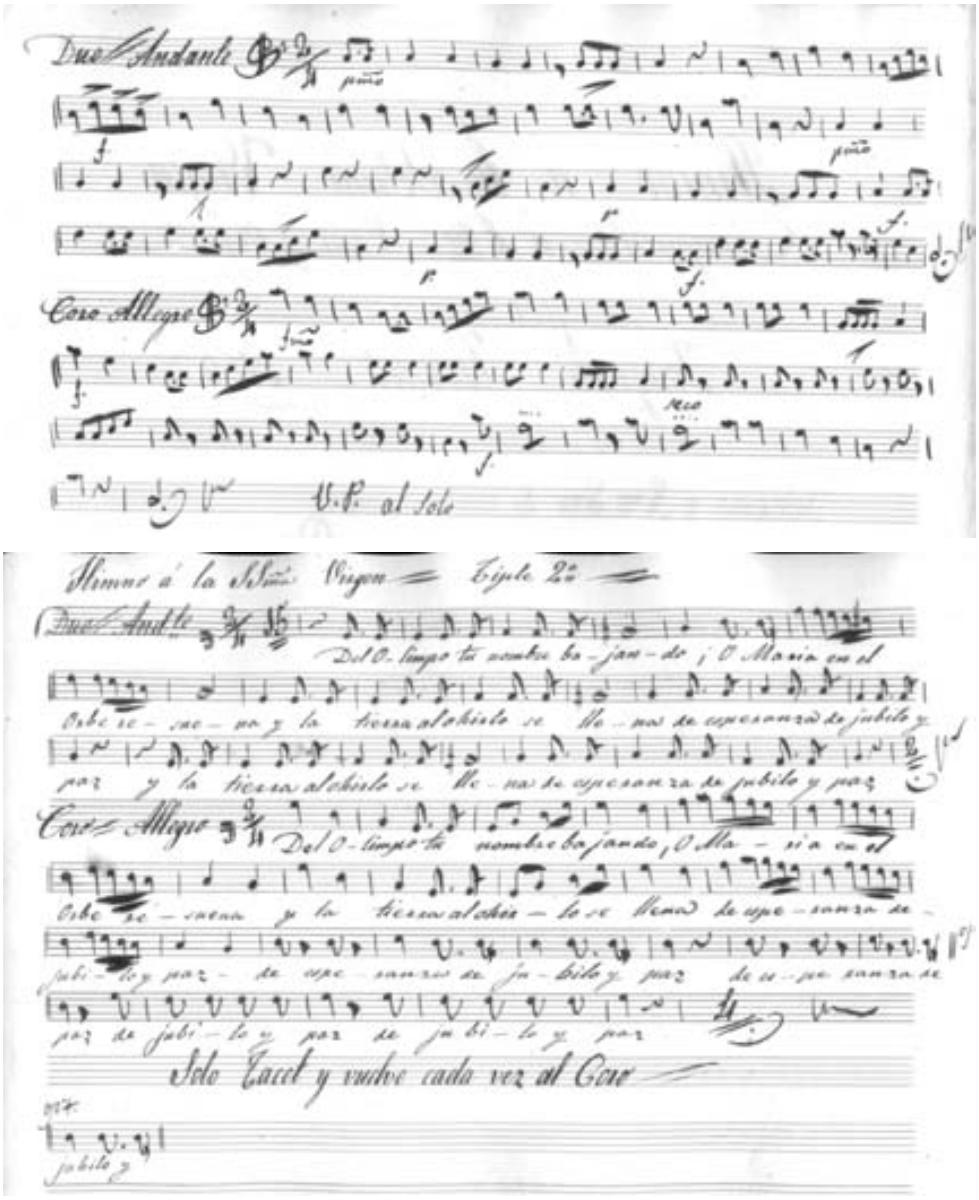


Figure A.1: Two examples of music scores of the composer Jovenet.

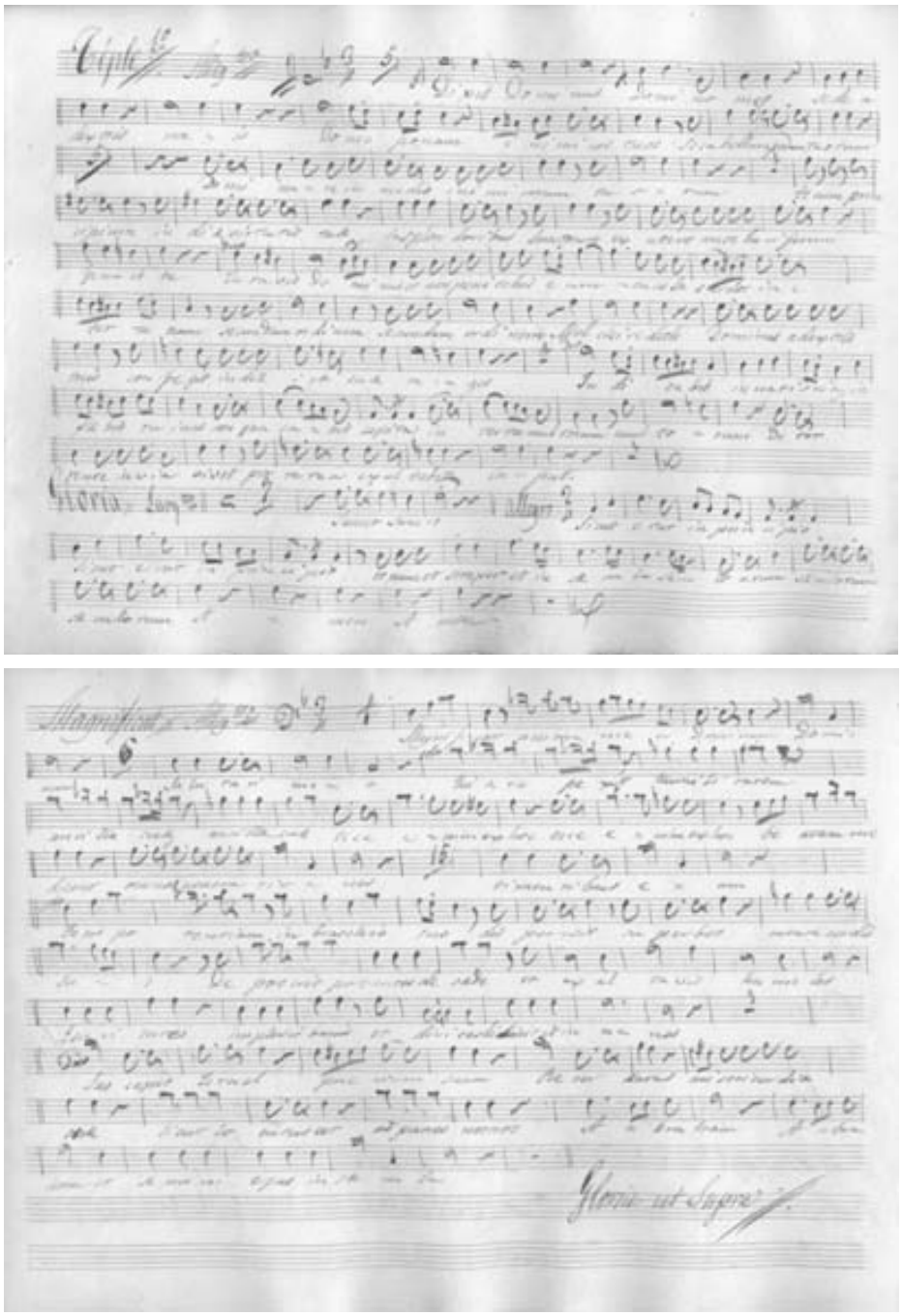


Figure A.2: Two examples of music scores of the composer Clausell.

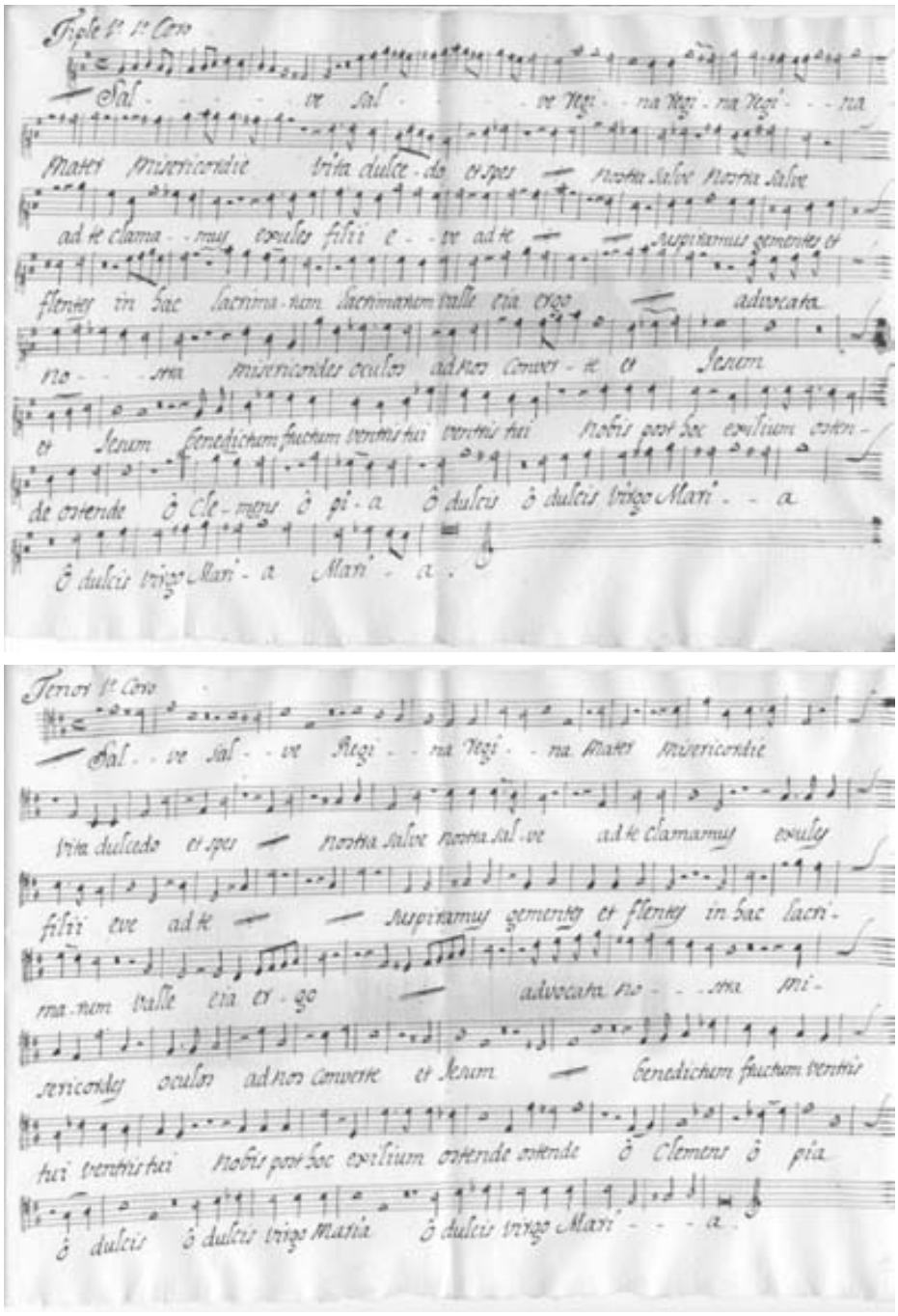


Figure A.3: Two examples of music scores of the composer Milans.

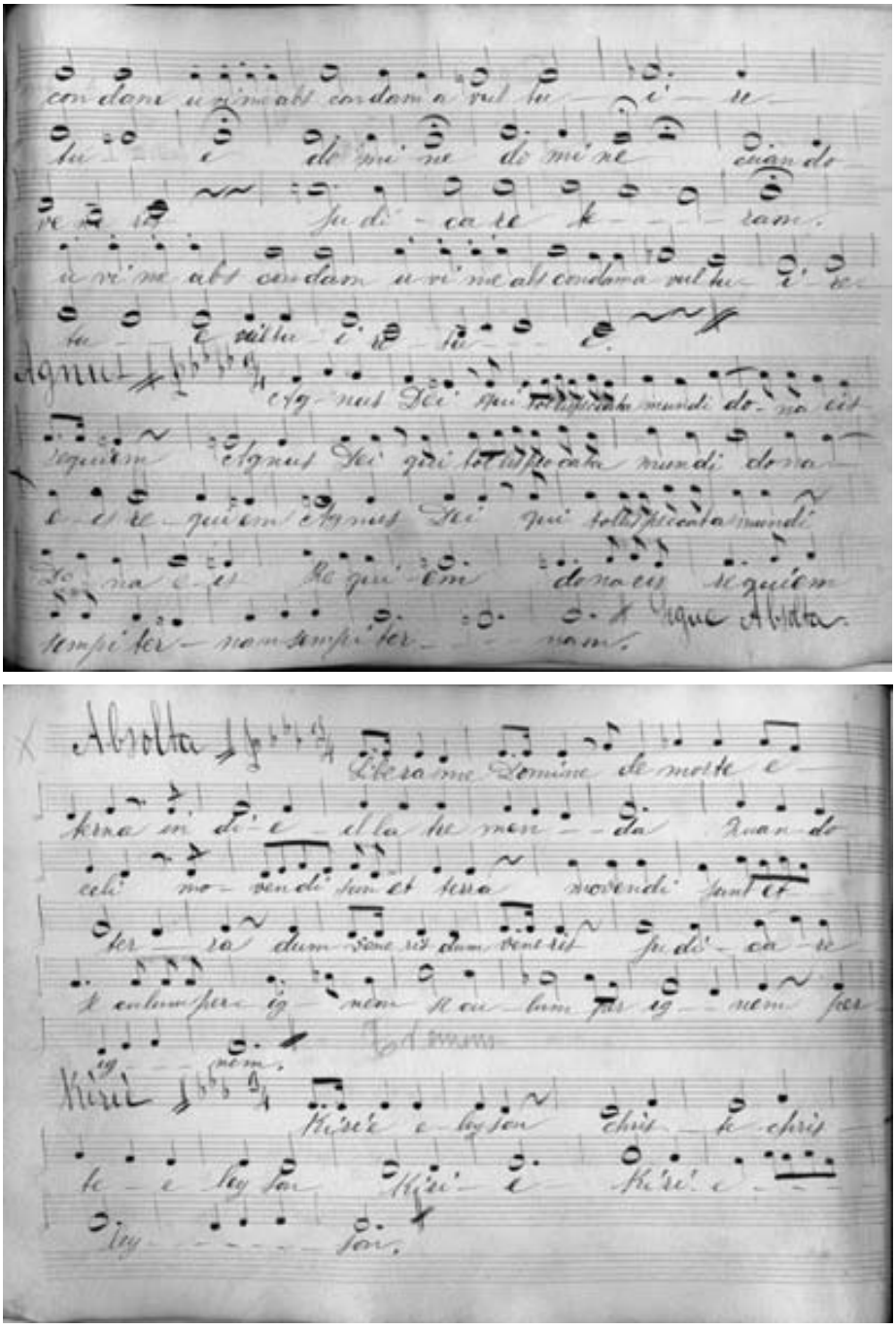


Figure A.4: Two examples of music scores of the composer Aleix.

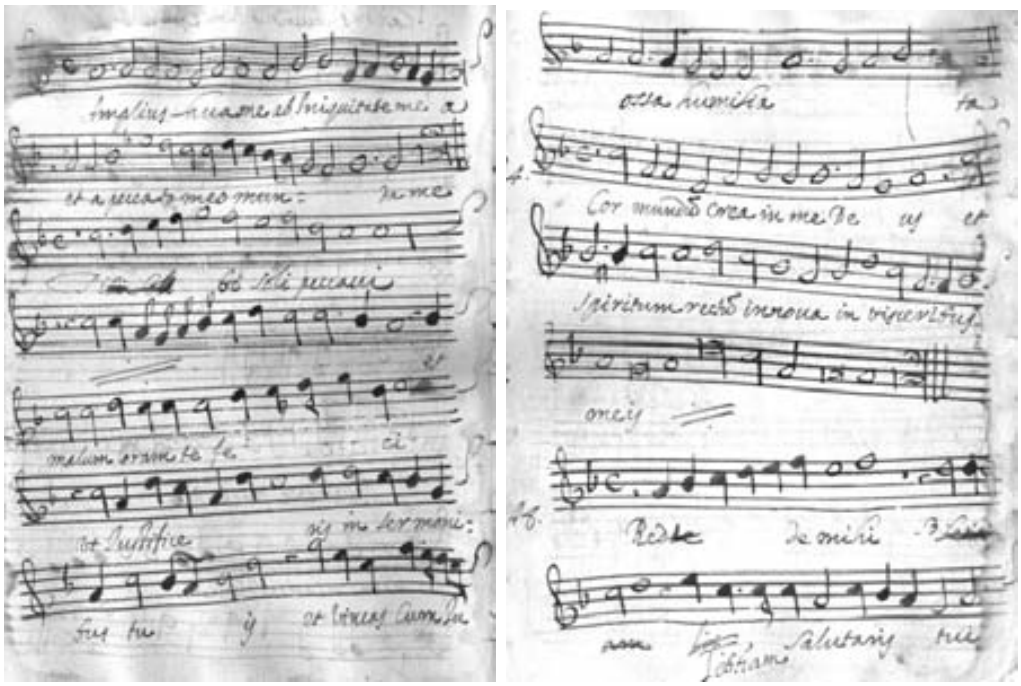


Figure A.5: Two examples of music scores of the composer Albareda.

A.2 Hand-drawn Music Symbols: Clefs and Accidentals

The database of hand-drawn music symbols has been created for validating the symbol recognition methods described in this work. It is composed of music clefs and accidentals, which have been manually segmented from a collection of modern and old music scores, and also, from a set of isolated music symbols drawn by different people. They have been scanned using the same scanner, at grey-scale and a resolution of 300 dpi, and stored as a bitmap file.

Music Clefs

The data set of music clefs contains 820 treble clefs, 549 bass clefs and 759 alto clefs. There are a total of 2128 instances from 22 different writers. The main characteristic of this dataset is that although there are only three different symbols, there is a high variability in the handwriting style. Figure A.6 shows some examples of treble clefs, Fig.A.7 shows bass clefs, and Fig.A.8 shows alto clefs. These clefs are written by different writers. Notice the high variability of the shape and sizes, mainly, alto clefs.

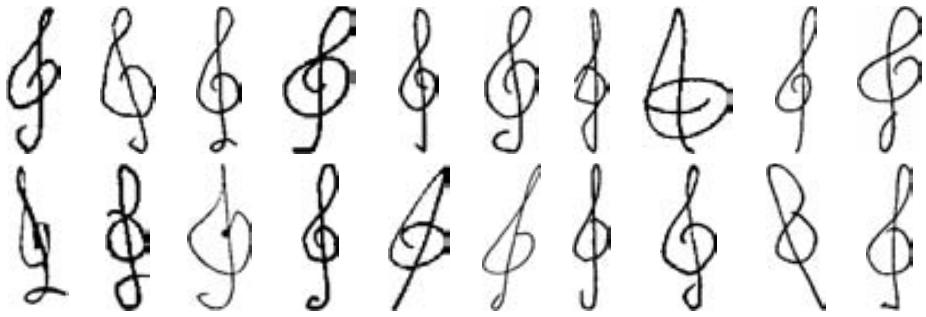


Figure A.6: Examples of treble clefs from different writers.

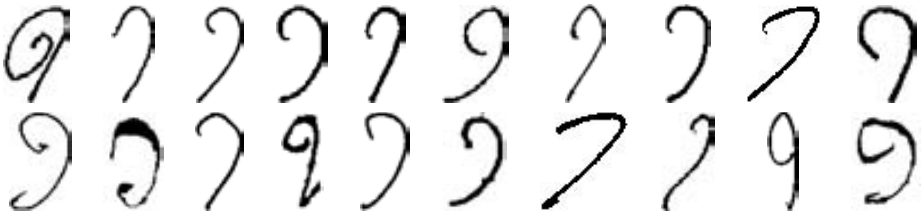


Figure A.7: Examples of bass clefs from different writers.

Figure A.8: Examples of alto clefs from different writers.

Music Accidentals

The data set of music accidentals contains 482 sharps, 472 naturals, 518 flats and 498 double sharps. There are a total of 1970 instances from 8 different writers. The main characteristic of this dataset is that accidentals are quite similar (see Fig.A.9).

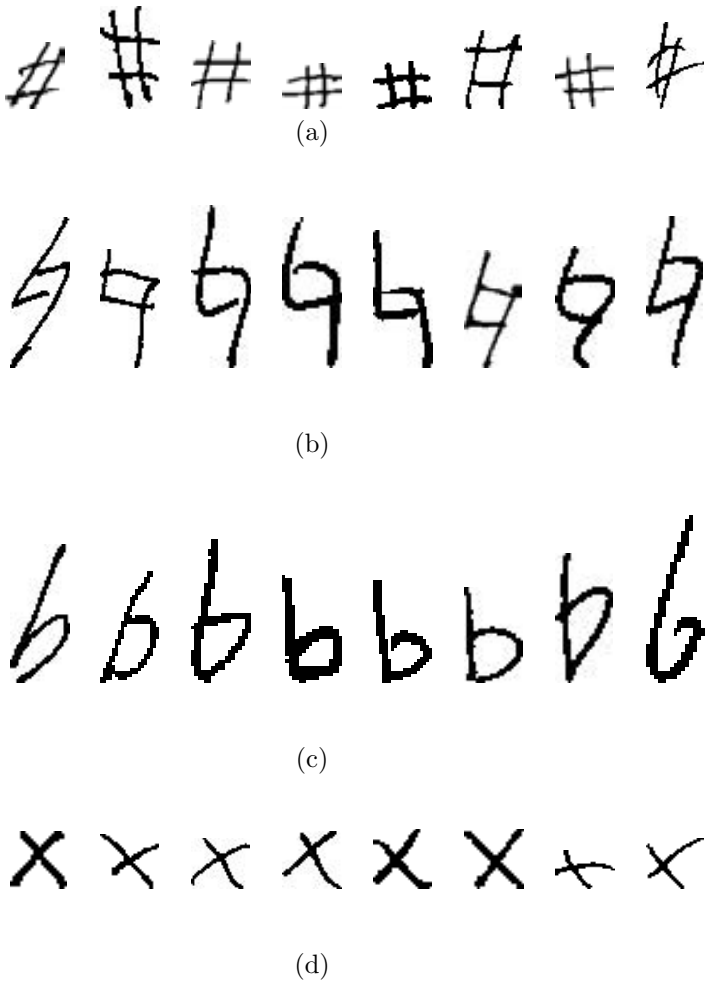


Figure A.9: Examples of accidentals from different writers. (a) Sharps, (b) Naturals, (c) Flats, (d) Double Sharps.

Appendix B

A Formal Grammar for Musical Scores Description

As it has been discussed in Section 9.2 (the future work) , context information can be formalized using a grammar to help in the recognition and classification tasks. The notation used for formalizing the grammar is shown in Table.B.1.

Notation	Meaning
|	or
[]	optional
$*$	repeat zero or more times
$+$	repeat one or more times

Table B.1: Notation used in the proposed grammar.

The proposed grammar for optical music recognition is the following.

<Score = <Heading with time signature> <Section> [<Final> <Heading> <Section>]* <Conclusive Ending>.

<Section> = <Measure> [<Bar line> <Measure>]*.

<Heading with time signature> = <clef> [<initial key signature>] <time signature>.

<Heading> = <clef> [<key signature>] [<time signature>].

<Final> = <double bar line> | <beggining repeat bar line>| <ending repeat bar line>.

<Conclusive Ending> = <double bar line> | <ending repeat bar line>.

<Clef> = <Treble> | <Alto> | <Bass>.

<Initial Key signature> = [♭]* | [♯]* .

<Key signature> = [♭]* [♮]* | [♯]* [♮]* | [♮]* [[♭]* | [♯]*].

<Time Signature> = 2/4 | 3/4 | 4/4 | C | 2/2 | 3/8 | 6/8 | 9/8 | 12/8 ...

<Measure> = [<Note> | <Rest>]$^{+}$.

<Note> = <White headnote> | <Headnote with a beam> | <beamed notes>.

<Rest> = <whole rest> | <half rest> | <quarter rest> | <eighth rest> | <sixteenth rest>.

<beamed notes> = [<headnote with a beam> <joining bar>]$^{+}$.

<Headnote with a beam> = <beam> <headnote> | <headnote> <beam>.

<headnote> = [accidental] <circle> <duration dot>.

<beam> = <vertical line> [<flag>]*.

<circle> = <white circle> | <filled circle>.

<accidental> = <♭> | <♮> | <♯> | <𝄫> | <×>.

<accidental> = <♭> | <♮> | <♯> | <𝄫> | <×>.

<duration dot> = [<dot>]$^{+}$.

Publications

Journal Papers

- Sánchez, G., Fornés, A., Mas, J., Lladós, J., *Herramientas de Visión por Computador para el aprendizaje de niños invidentes*, Novática, n.186, 33-38, March, 2007.
- Sánchez, G., Fornés, A., Mas, J., Lladós, J., *Computer Vision Tools for Visually Impaired Children Learning*, Upgrade, vol. VIII, n.2, 54-62, April, 2007.
- Fornés, A., Lladós, J., Sánchez, G. *Rotation Invariant Hand Drawn Symbol Recognition based on a Dynamic Time Warping*, Submitted to Pattern Recognition Letters.
- Escalera, S., Fornés, A., Pujol, O., Radeva, P., Sánchez, G., Lladós, J., *Blurred Shape Model for Binary and Grey-level Symbol Recognition.* Submitted to Pattern Recognition Letters (in Third Revision).
- Escalera, S., Fornés, A., Pujol, O., Lladós, J., Radeva, P., *Circular Blurred Shape Model for Multi-class Symbol Recognition.* Submited to Transactions on Systems, Man, and Cybernetics.

Book Chapters

- Fornés, A., Lladós, J., Sánchez, G. *Primitive Segmentation in Old Handwritten Music Scores*, Graphics Recognition: Ten Years Review and Future Perspectives, W. Liu, J. Lladós (Eds.), LNCS 3926: 288-299, January, 2006.
- Fornés, A., Escalera, S., Lladós, J., Sánchez, G., Radeva, P., Pujol, O. *Handwritten Symbol Recognition by a Boosted Blurred Shape Model with Error Correction.* 3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2007), J. Martí et al. (Eds.) LNCS 4477:13-21, Girona (Spain), June, 2007.
- Escalera, S., Fornés, A., Pujol, O., Lladós, J., Radeva, P. *Multi-class Binary Object Categorization using Blurred Shape Models.* Progress in Pattern Recognition, Image Analysis and Applications, 12th Iberoamerican Congress on Pattern (CIARP 2007), LCNS 4756:773-782, Valparaíso (Chile), November, 2007.

- Fornés, A., Lladós, J., Sánchez, G. *Old Handwritten Musical Symbol Classification by a Dynamic TimeWrapping Based Method.* Graphics Recognition: Recent Advances and New Opportunities, W. Liu, J. Lladós, J.M. Ogier (eds), LNCS 5046:52-60, July, 2008.

- Fornés, A., Escalera, S., Lladós, J., Sánchez, G., Mas, J. *Hand Drawn Symbol Recognition by Blurred Shape Model Descriptor and a Multiclass Classifier.* Graphics Recognition: Recent Advances and New Opportunities, W. Liu, J. Lladós, J.M. Ogier (eds), LNCS 5046:30-40, July, 2008.

- Valveny, E., Dosch, P., Fornés, A., Escalera, S. *Report on the Third Contest on Symbol Recognition.* Graphics Recognition: Recent Advances and New Opportunities, W. Liu, J. Lladós, J.M. Ogier (eds), LNCS 5046:321-328, July, 2008.

Conference and Workshop Contributions

- Fornés, A., Lladós, J., Sánchez, G. *Primitive Segmentation in Old Handwritten Music Scores.* 6th IAPR International Workshop on Graphics Recognition (GREC 2005), pp. 279-290, Hong Kong, Hong Kong SAR (China), August, 2005.

- Fornés, A., Lladós, J., Sánchez, G. *Staff and graphical primitive segmentation in old handwritten music scores.* Vuitè Congrés Català dIntel.ligència Artificial (CCIA), pp.83-90, Alghero, Italy, October, 2005.

- Fornés, A., Lladós, J., Sánchez, G. *Recognition of Old Handwritten Musical Scores.* 1st CVC Internal Workshop, Computer Vision: Progress of Research and Development, J. Lladós (ed.), pp.128-133, CVC (UAB), Bellaterra (Spain), October, 2006.

- Fornés, A., Escalera, S., Lladós, J., Sánchez, G., *Symbol Recognition by Multiclass Blurred Shape Models.* Seventh IAPR International Workshop on Graphics Recognition - GREC 2007 pp. 11-13, Curitiba (Brazil), September, 2007.

- Fornés, A., Lladós, J., Sánchez, G. *Old Handwritten Musical Symbol Classification by a Dynamic Time Warping Based Method.* Seventh IAPR International Workshop on Graphics Recognition - GREC 2007 pp. 26-27, Curitiba (Brazil), September, 2007.

- Fornés, A., Lladós, J., Sánchez, G. *A Dynamic Time Warping Based Method for Classifying Old Handwritten Musical Symbols.* Computer Vision: Advances in Research and Development, Proceedings of the 2nd CVC International Workshop, D. Gil, J. Gonzlez and G. Sánchez (eds.), pp. 24-27, Bellaterra (Spain), October, 2007.

- Fornés, A., Lladós, J., Sánchez, G., Bunke, H. *Writer Identification in Old Handwritten Music Scores.* Document Analysis system DAS 2008, Proceedings of the 8th International Workshop on Document Analysis Systems, pp. 347-353, Nara (Japan), September, 2008.

- Fornés, A., Lladós, J., Sánchez, G., Bunke, H. *On the writer identification in old handwritten music scores.* Current Challenges in Computer Vision. Proc. of the Third International Workshop (Eds. Benavente, Igual and Vilario), pp. 74-79, October, 2008.
- Fornés, A., Lladós, J., Sánchez, G., Bunke, H. *On the Use of Textural Features for Writer Identification in Old Handwritten Music Scores.* 10th International Conference on Document Analysis and Recognition. Preprint.
- Escalera, S., Fornés, A., G., Pujol, O, Radeva, P. *Multi-class Classification with Circular Blurred Shape Models.* 15th International Conference on Image Analysis and Processing (ICIAP). Preprint.
- Escalera, S., Fornés, A., G., Pujol, O, Escudero, A., Radeva, P. *Circular Blurred Shape Model for Symbol Spotting in Documents.* Submitted to International Conference on Image Processing (ICIP).

Technical Reports

- Fornés, A., *Analysis of Old Handwritten Musical Scores.* CVC Technical Report # 88, CVC (UAB), September, 2005.

Awards

- Best paper award of the 3rd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2007).
- Nomination to the best paper award of the *III Edición del Premio Novática al mejor artículo del año 2007.*

Bibliography

[AA82] A.Andronico and A.Ciampa. On automatic pattern recognition and adquisition of printed music. In *Proceedings of the International Computer Music Conference, ICDAR'95*, pages 245–278, Venice, Italy, August 1982.

[ABCdFC97] J.P. Antoine, D. Barachea, RM Cesar, and L. da Fontoura Costa. Shape characterization with the wavelet transform. *Signal Processing*, 62(3):265–290, 1997.

[ABS04] D. Anderson, C. Bailey, and M. Skubic. Hidden Markov Model Symbol Recognition for Sketch-Based Interfaces. AAAI 2004 Fall Symposium. In *Workshop on Making Pen-Based Interaction Intelligent and Natural,*, pages 15–21, Washington, DC, USA, Oct 2004.

[AC01] John Aach and George M. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508, 2001.

[AOC+00] S. Adam, JM Ogier, C. Cariou, R. Mullot, J. Labiche, and J. Gardes. Symbol and character recognition: application to engineering drawings. *International Journal on Document Analysis and Recognition*, 3(2):89–101, 2000.

[AST01] C. Ah-Soon and K. Tombre. Architectural symbol recognition using a network of constraints. *Pattern Recognition Letters*, 22(2):231 – 248, 2001.

[Bai04] H.S. Baird. Difficult and urgent open problems in document image analysis for libraries. *Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on*, pages 25–32, 2004.

[BB82] Dana H. Ballard and Christopher M. Brown. *Computer Vision*, chapter 4. Prentice Hall, 1982.

[BB92] Dorothea Blostein and Henry S. Baird. *Structured Document Image Analysis*, chapter A critical survey of music image analysis, pages 405–434. Springer Verlag, 1992.

[BB96] David Bainbridge and Tim Bell. An extensible optical music recognition system. In *Proceedings of the Nineteenth Australasian Computer Science Conference*, pages 308–317, Melbourne, 1996.

[BB08] R. Bertolami and H. Bunke. Hidden Markov model-based ensemble methods for offline handwritten text line recognition. *Pattern Recognition*, 41(11):3452–3460, 2008.

[BC97] David Bainbridge and Nicholas Carter. Automatic reading of music notation. *Handbook of Character recognition and document image analysis*, 24(8):583–603, 1997.

[BEVA06] S. Bres, V. Eglin, and C. Volpilhac-Auger. Evaluation of Handwriting Similarities Using Hermite Transform. In *Proceedings of 10th IWFHR*, pages 575–580, France, October 2006.

[BIM04] Ilvio Bruder, Temenushka Ignatova, and Lars Milewski. 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)*, pages 397–397, New York, NY, USA, 2004. ACM.

[BJFD98] Jose Luis Borbinha, Joaquim Jorge, Joao Ferreira, and Jose Delgado. A digital library for a virtual organization. *Hawaii International Conference on System Sciences*, 2:121, 1998.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[BPH03] A. Bensefia, T. Paquet, and L. Heutte. Information retrieval based writer identification. In *7th International Conference on Document Analysis and Recognition*, pages 946–950, Edinburgh, Scotland, UK, 3-6 August 2003.

[BPH05] Ameur Bensefia, Thierry Paquet, and Laurent Heutte. A writer identification and verification system. *Pattern Recognition Letters*, 26(13):2080–2092, 2005.

[BS06] M. Bulacu and L. Schomaker. Combining multiple features for text-independent writer identification and verification. In *Proc. of 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006)*, pages 281–286, 2006.

[BS07a] Marius Bulacu and Lambert Schomaker. Automatic handwriting identification on medieval documents. In Rita Cucchiara, editor, *International Conference on Image Analysis and Processing (ICIAP)*, pages 279–284. IEEE Computer Society, 2007.

[BS07b] Marius Bulacu and Lambert Schomaker. Text-independent writer identification and verification using textural and allographic features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(4):701–717, 2007.

[BSB07] Marius Bulacu, Lambert Schomaker, and A. Brink. Text-independent writer identification and verification on offline arabic handwriting. In *ICDAR*, pages 769–773. IEEE Computer Society, 2007.

[BSSS07] A. Bhardwaj, A. Singh, H. Srinivasan, and S. Srihari. On the use of Lexeme Features for writer verification. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 1088–1092, 2007.

[Bun82] H. Bunke. Attributed programmed graph grammars and their application to schematic diagram interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(6):574–582, 1982.

[Bun03] H. Bunke. Recognition of cursive Roman handwriting: past, present and future. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 448–459, 2003.

[BvKS+07] M. Bulacu, R. van Koert, L. Schomaker, T. van der Zant, et al. Layout Analysis of Handwritten Historical Documents for Searching the Archive of the Cabinet of the Dutch Queen. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 1-Volume 01*, pages 357–361. IEEE Computer Society Washington, DC, USA, 2007.

[BVSE97] V. Bouletreau, Nicole Vincent, Robert Sabourin, and Hubert Emptoz. Synthetic parameters for handwriting classification. In *Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR)*, pages 102–106, Washington, DC, USA, 1997. IEEE Computer Society.

[BVSE98] V. Bouletreau, N. Vincent, R. Sabourin, and H. Emptoz. Handwriting and signature: One or two personality identifiers? In *Proceedings of 14th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 1758–1760, 1998.

[BW97] H. Bunke and P.S.P. Wang. *Handbook of character recognition and document image analysis*. World Scientific Publishing, 1997.

[BYBKD07] I. Bar-Yosef, I. Beckman, K. Kedem, and I. Dinstein. Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents. *International Journal on Document Analysis and Recognition*, 9(2):89–99, 2007.

[BZB06] R. Bertolami, M. Zimmermann, and H. Bunke. Rejection strategies for offline handwritten text line recognition. *Pattern Recognition Letters*, 27(16):2005–2012, 2006.

[Car95] Nicholas P. Carter. Segmentation and preliminary recognition of madrigals notated in white mensural notation. *Machine Vision and Applications*, 5(3):223–230, 1995.

[CAVRPC+03] P.P. Cruz-Alcázar, E. Vidal-Ruiz, J.C. Pérez-Cortés, U.P. de Valencia, and S. Valencia. Musical style identification using grammatical inference: The encoding problem. *Lecture Notes in Computer Science, CIARP 2003*, 2905:375–382, 2003.

[CB92] Nicholas P. Carter and Richard A. Bacon. Automatic recognition of printed music. In H. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*, pages 169–203. Sringer-Verlag, 1992.

[CB08] D. Conklin and M. Bergeron. Feature set patterns in music. *Computer Music Journal*, 32(1):60–70, 2008.

[CBM88] A. Clarke, B.M. Brown, and M.P.Thorne. Inexpensive optical character recognition of music notation: a new alternative for publishers. In *Proceedings, Computers in Music Research Conference*, Bailrigg, Lancaster, UK, April 1988.

[CC01] M.T. Chang and S.Y. Chen. Deformed trademark retrieval based on 2D pseudo-hidden Markov model. *Pattern Recognition*, 34(5):953–967, 2001.

[CF06] J. Chapran and MC Fairhurst. Biometric writer identification based on the interdependency between static and dynamic features of handwriting. In *Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition*, pages 505–510, 2006.

[CF07] J. Chapran and M. Fairhurst. Automatic writer identification based on a velocity profile approach to processing of difficult handwriting styles. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 282–286, 2007.

[CGV+08] M. Coustaty, S. Guillas, M. Visani, K. Bertet, and J.M. Ogier. On the Joint Use of a Structural Signature and a Galois Lattice Classifier for Symbol Recognition. In W. Liu, J. Lladós, and J.M. Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities, Lecture Notes In Computer Science*, volume 5046, pages 61–70. Springer-Verlag Berlin, Heidelberg, 2008.

[Coü06] B. Coüasnon. DMOS, a generic document recognition method: application to table structure analysis in a general and in a specific way. *International Journal on Document Analysis and Recognition*, 8(2):111–122, 2006.

[CP06] M. Chan and J. Potter. Recognition of musically similar polyphonic music. In *18th International Conference on Pattern Recognition*

(ICPR 2006), volume 4, pages 809–812, Hong Kong, China, 20-24 August 2006.

[CPB+98] EG Caiani, A. Porta, G. Baselli, M. Turiel, S. Muzzupappa, F. Pieruzzi, C. Crema, A. Malliani, S. Cerutti, and D. di Bioingegneria. Warped-average template technique to track on a cycle-by-cyclebasis the cardiac filling phases on left ventricular volume. *Computers in Cardiology 1998*, pages 73–76, 1998.

[CR95] Bertrand Coüasnon and Bernard Rétif. Using a grammar for a reliable full score recognition system. In *International Computer Music Conference*, pages 187–194, Canada, 1995.

[CRH95] I. Cox, S. Roy, and SL Hingorani. Dynamic Histogram Warping of Images Pairs for Constant Image Brightness", IEEE Int. *Conf. on Image Processing*, 2:366–369, 1995.

[CS00a] S.H. Cha and S. Srihari. Multiple feature integration for writer verification. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 333–342, 2000.

[CS00b] Sung-Hyuk Cha and Sargur N. Srihari. Writer identification: Statistical analysis and dichotomizer. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 123–132, London, UK, 2000. Springer-Verlag.

[CS06] P. Clote and J. Straubhaar. Symmetric time warping, boltzmann pair probabilities and functional genomics. *Journal of Mathematical Biology*, 53(1):135–161, 2006.

[CS07] W. Chang and J. Shin. Modified Dynamic Time Warping for Stroke-Based On-line Signature Verification. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2-Volume 02*, pages 724–728. IEEE Computer Society Washington, DC, USA, 2007.

[CSE03] P. Clote, J. Straubhaar, and V. Ewell. An Application of Time Warping to Functional Genomics. Technical report, Technical Report, 2003.

[CW00] Yi-Kai Chen and Jhing-Fa Wang. Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1304–1317, 2000.

[CY00] K.F. Chan and D.Y. Yeung. Mathematical expression recognition: a survey. *International Journal on Document Analysis and Recognition*, 3(1):3–15, 2000.

[Dan98] Lee Sau Dan. Automatic optical music recognition, final year project report, 1998.

[DB95] TG Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *J. Artificial Intelligence Res.*, 2:263–286, 1995.

[DDPF08] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga. A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753, 2008.

[DJJ08] M. Delalandre, J.Lladós, and J.Ogier. A fast cbir system for old ornamental letter. In W. Liu, J. Lladós, and J.M. Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities, Lecture Notes In Computer Science*, volume 5046, pages 136–144. Springer-Verlag Berlin, Heidelberg, 2008.

[DL96] J.R. Davis and C. Lagoze. The networked computer science technical report library. *D-Lib Magazine*, 1996.

[DLDG+00] V. Di Lecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo, and A. Salzo. ZoningDesign for Hand-Written Numeral Recognition. In *Proceedings of the Seventh international Workshop on Frontiers in Handwriting Recognition–IWFHR*, volume 7, pages 583–588, Amsterdam, 2000.

[DMP08] Christoph Dalitz, Georgios K. Michalakis, and Christine Pranzas. Optical recognition of psaltic byzantine chant notation. *Int. J. Doc. Anal. Recognit.*, 11(3):143–158, 2008.

[DS82] I. Dinstein and Y. Shapira. Ancient hebraic handwriting identification with run-length histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 12(6):405–409, November 1982.

[Dun89] G.H. Dunteman. *Principal Components Analysis: Quantitative Application in Social Sciences*. Sage Publications, 1989.

[DV06] P. Dosch and E. Valveny. Report on the second symbol recognition contest. *Lecture Notes in Computer Science*, 3926:381, 2006.

[EBR07] V. Eglin, S. Bres, and C. Rivero. Hermite and Gabor transforms for noise reduction and handwriting classification in ancient manuscripts. *International Journal on Document Analysis and Recognition*, 9(2):101–122, 2007.

[ETP+08] S. Escalera, D.M.J. Tax, O. Pujol, P. Radeva, and R.P.W. Duin. Subclass problem-dependent design for error-correcting output codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1041–1054, June 2008.

[FAT05] MA Ferrer, JB Alonso, and CM Travieso. Offline geometric parameters for automatic signature verification using fixed-point arithmetic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):993–997, 2005.

[FB93] Hoda Fahmy and Dorothea Blostein. *Machine Vision and Applications*, chapter A graph grammar programming style for recognition of music notation, pages 83–99. Springer Verlag, 1993.

[FBS02] K. Franke, O. Bunnemeyer, and T. Sy. Ink texture analysis for writer identification. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 268–273, 2002.

[FHT00] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–374, 2000.

[FJ03] M.J. Fonseca and J.A. Jorge. Towards content-based retrieval of technical drawings through high-dimensional indexing. *Computers & Graphics*, 27(1):61–69, 2003.

[FMKK00] P. Franti, A. Mednonogov, V. Kyrki, and H. Kalviainen. Content-based matching of line-drawing images using the Hough transform. *International Journal on Document Analysis and Recognition*, 3(2):117–124, 2000.

[FPJ02] M.J. Fonseca, C. Pimentel, and J.A. Jorge. CALI: An Online Scribble Recognizer for Calligraphic Interfaces. In *AAAI Spring Symposium on Sketch Understanding*, pages 51–58, 2002.

[Fre61] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput (EC)*, 10(2):260–268, 1961.

[Fuj04] I. Fujinaga. Staff detection and removal. In S. George, editor, *Visual Perception of Music Notation*, pages 1–39. Idea Group, 2004.

[GÖ3] Roland Göcke. 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)*, pages 250–255, Rhodes, Greece, 30 June – 2 July 2003.

[GB04] S. Günter and H. Bunke. HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components. *Pattern Recognition*, 37(10):2069–2079, 2004.

[GBA07] S. Gazzah and N.E. Ben Amara. Arabic handwriting texture analysis for writer identification using the dwt-lifting scheme. *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, 2:1133–1137, Sept. 2007.

[GC04] U. Garain and BB Chaudhuri. Recognition of Online Handwritten Mathematical Expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(6):2366–2376, 2004.

[GD95] D.M. Gavrila and L.S. Davis. Towards 3-D Model-based Tracking and Recognition of Human Movement. In Martin Bichsel, editor, *Int. Workshop on Face and Gesture Recognition*, pages 272–277, June 1995.

[Gha01] R. Ghani. Combining labeled and unlabeled data for text classification with a large number of categories. In *Proceedings of the IEEE International Conference on Data Mining*, volume 2, 2001.

[GL96] Jonas Garding and Tony Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *International Journal of Computer Vision*, 17(2):163–191, February 1996.

[GNP+04] Basilios Gatos, Kostas Ntzios, Ioannis Pratikakis, Sergios Petridis, T. Konidaris, and Stavros J. Perantonis. A segmentation-free recognition technique to assist old greek handwritten manuscript ocr. In *Proceedings of 6th international workshop of Document Image Analysis, DAS 2004*, pages 63–74, Italy, 2004.

[GP95] K. Gollmer and C. Posten. Detection of distorted pattern using dynamic time warping algorithm and application for supervision of bioprocesses. *On-Line Fault Detection and Supervision in Chemical Process Industries*, 1995.

[GSTL02] J. Gu, H. Z. Shu, C. Toumoulin, and L. M. Luo. A novel algorithm for fast computation of zernike moments. *Pattern Recognition*, 35(12):2905 – 2911, 2002.

[Har79] R.M Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67:786–804, 1979.

[HB03] Caroline Hertel and Horst Bunke. A set of novel features for writer identification. In *Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 679–687, 2003.

[HDT03] N. Hu, RB Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–188, New Paltz, New York, October 2003.

[HL06] Wladyslaw Homenda and Marcin Luckner. Automatic knowledge acquisition: Recognizing music notation with methods of centroids and classifications trees. In *IJCNN*, pages 3382–3388. IEEE, 2006.

[HN04] H. Hse and AR Newton. Sketched symbol recognition using Zernike moments. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 367–370, 2004.

[Hom05] Wladyslaw Homenda. Optical music recognition: the case study of pattern recognition. In Marek Kurzynski, Edward Puchala, Michal

Wozniak, and Andrzej Zolnierek, editors, *CORES*, volume 30 of *Advances in Soft Computing*, pages 835–842. Springer, 2005.

[HS08] PS Hiremath and S. Shivashankar. Wavelet based co-occurrence histogram features for texture classification with an application to script identification in a document image. *Pattern Recognition Letters*, 29(9):1182–1189, 2008.

[HSD73] R.M Haralick, K. Shnmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 3(6):610–621, 1973.

[HYT+06] Z. He, X. You, Y.Y. Tang, B. Fang, and J. Du. Handwriting-based personal identification. *International Journal of Pattern Recognition and Artificial Intelligence*, 20(2):209, 2006.

[HZW08] G. Huang, W. Zhang, and L. Wenyin. A discriminative representation for symbolic image similarity evaluation. In W. Liu, J. Lladós, and J.M. Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities, Lecture Notes In Computer Science*, volume 5046, pages 71–79. Springer-Verlag Berlin, Heidelberg, 2008.

[JME+07] N. Journet, R. Mullot, V. Eglin, J.Y. Ramel, and T. LI. A proposition of retrieval tools for historical document images libraries. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 1053–1057, 2007.

[KA05] L. Kotoulas and I. Andreadis. Real-time computation of Zernike moments. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(6):801–809, 2005.

[KCKK04] Min Soo Kim, Kyu Tae Cho, Hee Kue Kwag, and Jin Hyung Kim. Segmentation of handwritten characters for digitalizing korean historical documents. In *Proceedings of 6th international workshop of Document Image Analysis, DAS 2004*, Italy, 2004.

[KFK02] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis. An unconstrained handwriting recognition system. *International Journal on Document Analysis and Recognition*, 4(4):226–242, 2002.

[KGWM01] J. Kittler, R. Ghaderi, T. Windeatt, and J. Matas. Face verification using error correcting output codes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 755–760. IEEE Computer Society; 1999, 2001.

[KH90] A. Khotanzad and YH Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[KI91] Hirokazu Kato and Seiji Inokuchi. *Structured Document Image Analysis*, chapter A recognition system for printed piano music using musical knowledge and constraints, pages 435–455. Springer-Verlag, 1991.

[Kit78] J. Kittler. Feature set search algorithms. In C.H.Chen, editor, *Pattern Recognition and Signal Processing*, 1978.

[KK99] WY Kim and YS Kim. A new region-based shape descriptor. Technical report, Hanyang University and Konan Technology, 1999.

[KL83] Joseph B. Kruskal and Mark Liberman. The symmetric time-warping problem: From continuous to discrete. In David Sankoff and Joseph B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 125–161, Reading, Massachusetts, September 1983. Addison-Wesley Publishing Co.

[KMA04] E.M. Kornfield, R. Manmatha, and J. Allan. Text alignment with handwritten documents. In *Document Image Analysis for Libraries*, pages 195–209, Washington, DC, USA, 2004. IEEE Computer Society.

[KP99] Eamonn Keogh and M. Pazzani. Scaling up dynamic time warping to massive datasets. In J. M. Zytkow and J. Rauch, editors, *3rd European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'99)*, volume 1704, pages 1–11, Prague, Czech Republic, 1999. Springer.

[KR05] E. Keogh and C.A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[KSK93] SH Kim, JW Suh, and JH Kim. Recognition of logic diagrams by identifying loops and rectilinearpolylines. In *Document Analysis and Recognition, Proceedings of the Second International Conference on*, pages 349–352, 1993.

[KUKO08] A. Karray, S. Uttama, S. Kanoun, and J. Ogier. An ancient graphic documents indexing method based on spatial similarity. In W. Liu, J. Lladós, and J.M. Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities, Lecture Notes In Computer Science*, volume 5046, pages 126–134. Springer-Verlag Berlin, Heidelberg, 2008.

[Kun04] L.I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, 2004.

[KV00] Z.M. Kovács-Vajna. A fingerprint verification system based on triangular matching and dynamic time warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1266–1276, 2000.

[LC85] Myung Woo Lee and Jong Soo Choi. The recogniton of printed music score and performance using computer vision system. *Journal of the Korea Institute of Electronic Engineers*, 22(5):429–435, setember 1985.

[LC03] G. Leedham and S. Chachra. Writer identification using innovative binarised features of handwritten numerals. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 413–416, 2003.

[LCL93] I. Leplumey, J. Camillerapp, and G. Lorette. A robust detector for music staves. In *Proceedings of the International Conference on Document Analysis and Recognition*, Japan, 1993.

[Les97] M. Lesk. *Practical Digital Libraries: Books, Bytes, and Bucks*. Morgan Kaufmann, 1997.

[LG02] J. Liu and P. Gader. Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition. *Pattern Recognition*, 35(10):2061–2071, 2002.

[Liu07] C.L. Liu. Normalization-Cooperated Gradient Feature Extraction for Handwritten Character Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1465–1469, 2007.

[LMV01] Josep Lladós, Enric Martí, and Juan José Villanueva. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 23(10), pages 1137–1143, October 2001.

[Log04] Logitech. I(o) digital pen : http://www.logitech.com, 2004.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LP92] E. Levin and R. Pieraccini. Dynamic planar warping for optical character recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 149–152, 1992.

[LP94] F. Leclerc and R. Plamondon. Automatic signature verification: The state of the art. *International Journal of Pattern Recognition and Artificial Intelligence (PRAI)*, 8(3):643–660, 1994.

[LS07] J. Llados and G. Sanchez. Indexing Historical Documents by Word Shape Signatures. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 362–366, 2007.

[LSL05] S. Liang, Z. Sun, and B. Li. Sketch Retrieval Based on Spatial Relations. *Proceedings of International Conference on Computer Graphics, Imaging and Visualization, Beijing, China*, pages 24–29, 2005.

[LSS07] L. Likforman-Sulem and M. Sigelle. Recognition of broken characters from historical printed books using dynamic bayesian networks. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 173–177, 2007.

[Lut02] Nailja Luth. Automatic identification of music notations. In *Proceedings of the Second International Conference on WEB Delivering of Music (WEDELMUSIC)*, 2002.

[LVSM02] J. Lladós, E. Valveny, G. Sánchez, and E. Martí. Symbol Recognition: Current Advances and Perspectives. In *Lecture Notes in Computer Science, vol. 2390*, pages 104–128. Springer, 2002.

[LWD06] X. Li, X. Wang, and X. Ding. An off-line Chinese writer retrieval system based on text-sensitive writer identification. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, 2006.

[LWJ04] Y. Lin, L. Wenyin, and C. Jiang. A structural approach to recognizing incomplete graphic objects. In *Pattern Recognition. Proceedings of the 17th International Conference on*, volume 1, pages 371–375. IEEE Computer Society Washington, DC, USA, 2004.

[Mah82] J.V. Mahoney. Automatic analysis of musical score images, b.s. thesis, 1982.

[MB96] B.T. Messmer and H. Bunke. Automatic learning and recognition of graphical symbols in engineering drawings. *Lecture Notes In Computer Science*, 1072:123–134, 1996.

[MB98] BT Messmer and H. Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):493–504, 1998.

[MB01] U.V. Marti and H. Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[MCC08] I. Martinat, B. Couasnon, and J. Camillerapp. An Adaptative Recognition System using a Table Description Language for Hierarchical Table Structures in Archival Documents. In W. Liu, J. Lladós, and J.M. Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities, Lecture Notes In Computer Science*, volume 5046, pages 9–20. Springer-Verlag Berlin, Heidelberg, 2008.

[MCF00] N. Meyyappan, GG Chowdhury, and S. Foo. A Review Of The Status Of Twenty Digital Libraries. *Journal of Information Science*, 26(5):337–355, 2000.

[MG99] S. Madhvanath and V. Govindaraju. Local reference lines for handwritten phrase recognition. *Pattern Recognition*, 32(12):2021–2028, 1999.

[MJSL08] J Mas, J.A. Jorge, G. Sánchez, and J. Lladós. Representing and parsing sketched symbols using adjacency grammars and a grid-directed parser. In J.M. Ogier eds. W. Liu, J. Lladós, editor, *Graphics Recognition: Recent Advances and New Opportunities, Lecture Notes in Computer Science*, volume 5046, pages 176–187. Springer-Verlag, 2008.

[MKJ08] Y. Mingqiang, K. Kidiyo, and R. Joseph. A Survey of Shape Feature Extraction Techniques. In Peng-Yeng Yin, editor, *Pattern Recognition Techniques, Technology and Applications*, pages 43–90. I-Tech, Vienna, Austria, November 2008.

[MKL97] Babu M. Mehtre, Mohan S. Kankanhalli, and Wing F. Lee. Shape measures for content based image retrieval: A comparison. *Information Processing & Management*, 33(3):319–337, May 1997.

[MM86] F. Mokhtarian and A.K. Mackworth. Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):34–43, 1986.

[MM07] H. Miyao and M. Maruyama. An online handwritten music symbol recognition system. *International Journal on Document Analysis and Recognition*, 9(1):49–58, 2007.

[MMB01] U.V. Marti, R. Messerli, and H. Bunke. Writer identification using text line based features. In *Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR)*, pages 101–105, 2001.

[MN95] Hidetoshi Miyao and Yasuaki Nakano. Head and stem extraction from printed music scores using a neural network approach. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, pages 1074–1079, 1995.

[MNG07] M. Makridis, N. Nikolaou, and B. Gatos. An Efficient Word Segmentation Technique for Historical and Degraded Machine-Printed Documents. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 178–182, 2007.

[MP99] M.E. Munich and P. Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proc. of the 8 thIEEE International Conference on Computer Vision*, pages 108–115, Korfu, Greece, 1999.

[MPE] MPEG7. Mpeg7 repository database: http://knight.cis.temple.edu/ shape/mpeg7/dataset.html.

[MR00] S. Müller and G. Rigoll. Engineering Drawing Database Retrieval Using Statistical Pattern Spotting Techniques. *Graphics Recognition-Recent Advances, Lecture Notes in Computer Science*, 1941:246–255, 2000.

[MRHS93] Bharath R. Modayur, Visvanathan Ramesh, Robert M. Haralick, and Linda G. Shapiro. Muser: A prototype musical score recognition system using mathematical morphology. *Machine Vision and Applications*, 6(2-3):140–150, 1993.

[MSH89] T. Matsushima, S.Ohteru, and S. Hashimoto. An integrated music information processing system: Psb-er. In *In proceedings of 1989 International Computer Music Conference*, pages 191–198, Columbus, Ohio, November 1989.

[MSS02] BS Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley, 2002.

[MTM07] C. Mancas-Thillou and M. Mancas. Comparison between pen-scanner and digital camera acquisition for engraved character recognition. In Koichi Kise and David S. Doermann, editors, *Proc. of the 2nd. International Workshop on Camera-Based Document Analysis and Recognition*, volume 130-137, Curitiba, Brazil, September 2007.

[Ng01] Kia Ng. Music manuscript tracing. In *International Workshop on Graphics Recognition Algorithms and Applications, GREC 2001*, Japan, September 2001.

[Nib86] W. Niblack. *An Introduction to Digital Image Processing*. Prentice Hall, 1986.

[Nie04] R. Niels. Dynamic Time Warping: An intuitive way of handwriting recognition. Master's thesis, Radboud University Nijmegen, The Netherlands, November/December 2004.

[NV05] R. Niels and L. Vuurpijl. Using Dynamic Time Warping for intuitive handwriting recognition. *Advances in Graphonomics, Proceedings of the 12th Conference of the International Graphonomics Society*, pages 217–221, 2005.

[OFC99] T. Oates, L. Firoiu, and P.R. Cohen. Clustering Time Series with Hidden Markov Models and Dynamic Time Warping. *Proceedings of the IJCAI Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning*, pages 17–21, 1999.

[OS01] Nicola Orio and Diemo Schwarz. Alignment of monophonic and polyphonic music to a score. In San Francisco International Computer

Music Association, editor, *Proceedings of the International Computer Music Conference*, pages 155–158, Havana, Cuba, September 2001.

[OSU] OSU. Osu-svm-toolbox: http://svm.sourceforge.net/.

[PBF07] Laurent Pugin, John Ashley Burgoyne, and Ichiro Fujinaga. Goal-directed evaluation for the improvement of optical music recognition on early music prints. In Edie M. Rasmussen, Ray R. Larson, Elaine Toms, and Shigeo Sugimoto, editors, *JCDL*, pages 303–304. ACM, 2007.

[PBS+07] L. Prasanth, V. Babu, R. Sharma, GV Rao, and M. Dinesh. Elastic matching of online handwritten tamil and telugu scripts using local features. In *Document Analysis and Recognition (ICDAR). Ninth International Conference on*, volume 2, pages 1028–1032, 2007.

[PER08] O. Pujol, S. Escalera, and P. Radeva. An incremental node embedding technique for error correcting output codes. *Pattern Recognition*, 41(2):713–725, 2008.

[PHBF08] Laurent Pugin, Jason Hockman, John Ashley Burgoyne, and Ichiro Fujinaga. GAMERA versus ARUSPIX. Two Optical Music Recognition Approaches. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 419–424. Lulu. com, 2008.

[PJJ94] P.Pudil, J.Novovicová, and J.Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[PL89] R. Plamondon and G. Lorette. Automatic signature verification and writer identification: The state of the art. *PR*, 22(2):107–131, 1989.

[PLWH04] B. Peng, Y. Liu, L. Wenyin, and G. Huang. Sketch Recognition Based on Topological Spatial Relationship. In *Structural, Syntactic, and Statistical Pattern Recognition: Lecture Notes in Computer Science*, volume 3138, pages 434–443. Springer-Verlag, 2004.

[PPR00] J. Parker, J. Pivovarov, and D. Royko. Vector Templates for Symbol Recognition. In *International Conference on Pattern Recognition*, volume 15, pages 602–605, 2000.

[Pre70] D.S. Prerau. Computer pattern recognition of standard engraved music notation, phd thesis, 1970.

[PRT] PRTools. Prtools toolbox - faculty of applied physics, delft university of technology, the netherlands: http://www.prtools.org/.

[Pru66] D.H. Pruslin. Automatic recognition of sheet music, phd thesis, 1966.

[PS00] R. Plamondon and S.N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, Jan 2000.

[PT97] G.S. Peake and T.N. Tan. Script and language identification from document images. *Document Image Analysis, 1997. (DIA '97) Proceedings., Workshop on*, pages 10–17, Jun 1997.

[Pug06] Laurent Pugin. Optical music recognitoin of early typographic prints using hidden markov models. In *ISMIR*, pages 53–56, 2006.

[PVS03] J.C. Pinto, P. Vieira, and J.M. Sosa. A new graph-like classification method applied to ancient handwritten musical symbols. *International Journal of Document Analysis and Recognition (IJDAR)*, 6(1):10–22, 2003.

[Rab89] LR Rabiner. A tutorial on hidden Markov models and selected applications inspeech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RB05] F. Rossant and I. Bloch. Optical music recognition based on a fuzzy modeling of symbol classes and music writing rules. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages 538–541, Sept. 2005.

[RBO08] R. Raveaux, J.C. Burie, and J.M. Ogier. A Segmentation Scheme Based on a Multi-graph Representation: Application to Colour Cadastral Maps. In W. Liu, J. Lladós, and J.M. Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities, Lecture Notes In Computer Science*, volume 5046, pages 202–212. Springer-Verlag Berlin, Heidelberg, 2008.

[RCF+93] R. Randriamahefa, J.P. Cocquerez, C. Fluhr, F. Pépin, and S. Philipp. Printed music recognition. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 898–901, Japan, 1993.

[RF03] J. Riley and I. Fujinaga. Recommended best practices for digital image capture of musical scores. *OCLC Systems and Services*, 19(2):62–69, 2003.

[RJ93] L. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1993.

[RK05] C.A. Ratanamahatana and E. Keogh. Three myths about Dynamic Time Warping Data Mining. *Proceedings of SIAM International Conference on Data Mining (SDM'05)*, 2005.

[RLD07] M. Rusinol, J. Llados, and P. Dosch. Camera-Based Graphical Symbol Detection. In *Document Analysis and Recognition (ICDAR). Ninth International Conference on*, volume 2, pages 884–888, September 2007.

[RM02] Toni M. Rath and R. Manmatha. Lower-bounding of dynamic time warping distances for multivariate time series. Technical report, Technical Report MM-40, Center for Intelligent Information Retrieval, University of Massachusetts Amherst, 2002.

[RM03] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 521–527. Madison, WI, June 18-20 2003.

[RnL08] Marçal Rusiñol and Josep Lladós. A region-based hashing approach for symbol spotting in technical documents. In *Graphics Recognition. Recent Advances and New Opportunities: 7th International Workshop, GREC 2007, Curitiba, Brazil, September 20-21, 2007. Selected Papers*, pages 104–113, Berlin, Heidelberg, 2008. Springer-Verlag.

[RnLS09] Marçal Rusiñol, Josep Lladós, and Gemma Sánchez. Symbol spotting in vectorized technical drawings through a lookup table of region strings. *Pattern Analysis & Applications*, 2009.

[RPD01] R.O.Duda, P.E.Hart, and D.G.Stork. *Pattern Classification*. Wiley Interscience, 2001.

[RRC07] C. Renaudin, Y. Ricquebourg, and J. Camillerapp. A general method of segmentation-recognition collaboration applied to pairs of touching and overlapping symbols. In *Document Analysis and Recognition (ICDAR). Ninth International Conference on*, volume 2, pages 659–663, 2007.

[RT88] J.W. Roach and J.E. Tatem. Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment. In *In Proceedings of Pattern Recongition*, volume 21(1), pages 33–44, 1988.

[RTV07] O Ramos, S. Tabbone, and E. Valveny. A review of shape descriptors for document analysis. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 227–231, September 2007.

[SAP+06] K.P. Sankar, V. Ambati, L. Pratha, CV Jawahar, and I. Hyderabad. Digitizing a million books: Challenges for document analysis. *Lecture Notes in Computer Science*, 3872:425, 2006.

[SB04] L. Schomaker and M. Bulacu. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):787–798, June 2004.

[SB07a] A. Schlapbach and H. Bunke. A writer identification and verification system using HMM based recognizers. *Pattern Analysis & Applications*, 10(1):33–43, 2007.

[SB07b] A. Schlapbach and H. Bunke. Fusing Asynchronous Feature Streams for On-line Writer Identification. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 103–107, 2007.

[SB08] Andreas Schlapbach and Horst Bunke. Off-line writer identification and verification using gaussian mixture models. In Simone Marinai and Hiromichi Fujisawa, editors, *Machine Learning in Document Analysis and Recognition*, volume 90 of *Studies in Computational Intelligence*, pages 409–428. Springer, 2008.

[SBB+05] SN Srihari, MJ Beal, K. Bandi, V. Shah, and P. Krishnamurthy. A statistical model for writer verification. In *Document Analysis and Recognition. Proceedings. Eighth International Conference on*, pages 1105–1109, 2005.

[SCAL02] S.N. Srihari, S.H. Cha, H. Arora, and S. Lee. Individuality of handwriting. *Journal of Forensic Sciences*, 47(4):856–872, 2002.

[Sch07a] A. Schlapbach. *Writer Identification and Verification*. PhD thesis, University of Bern, 2007.

[Sch07b] L. Schomaker. Advances in Writer identification and verification. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 1268–1273, 2007.

[SD99] Marc Vuilleumier Stückelberg and David S. Doermann. On musical score recognition using probabilistic reasoning. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 115–118, 1999.

[SG07] Z. Saidane and C. Garcia. Automatic scene text recognition using a convolutional neural network. In Koichi Kise and David S. Doermann, editors, *Proc. of the 2nd. International Workshop on Camera-Based Document Analysis and Recognition*, volume 100-106, Curitiba, Brazil, September 2007.

[SGV03] A. Seropian, M. Grimaldi, and N. Vincent. Writer Identification based on the fractal construction of a reference base. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 1163–1167, 2003.

[SKB05] A. Schlapbach, V. Kilchherr, and H. Bunke. Improving writer identification by means of feature selection and extraction. In *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR)*, pages I: 131–135, 2005.

[SLS07] Yu Shi, HaiYang Li, and F.K. Soong. A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. *Ninth International Conference on Document Analysis and Recognition*, 2:854–858, Sept. 2007.

[SM92] F. Stein and G. Medioni. Structural indexing: Efficient 3-D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.

[SM99] T. Syeda-Mahmood. Indexing of Technical Line Drawing Databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 737–751, 1999.

[SOC99] M. Schmill, T. Oates, and P. Cohen. Learned models for continuous planning. In *In Proceedings of Uncertainty 99: The Seventh In Proceedings of Uncertainty 99: The Seventh International Workshop on Articial Intelligence and Statistics*, pages 278–282, Fort Lauderdale, Florida, USA, January 1999.

[Spi04] A. Lawrence Spitz. Tilting at windmills: adventures in attempting to reconstruct don quixote. In *Proceedings of 6th international workshop of Document Image Analysis, DAS 2004*, Italy, 2004.

[SRS03] F. Soulez, X. Rodet, and D. Schwarz. Improving polyphonic and poly-instrumental music to score alignment. *International Conference on Music Information Retrieval, Baltimore*, 2003.

[SS96] H. Samet and A. Soffer. MARCO: MAp retrieval by COntent. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):783–798, 1996.

[STB00] H.E.S. Said, T.N. Tan, and K.D. Baker. Personal identification based on handwriting. *Pattern Recognition*, 33(1):149–160, January 2000.

[Suw05] M. Suwa. Segmentation of connected handwritten numerals by graph representation. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 750–754, 2005.

[SV07] IA Siddiqi and N. Vincent. Writer Identification in Handwritten Documents. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 108–112, 2007.

[SVL+04] G. Sanchez, E. Valveny, J. Llados, J. Mas Romeu, and N. Lozano. A platform to extract knowledge from graphic documents. application to an architectural sketch understanding scenario. In A. Dengel S. Marinai, editor, *Document Analysis Systems VI, Lecture Notes in Computer Science*, volume 3163, pages 389–400, Florence - Italy, 2004. Springer-Verlag.

[SW08] J.P. Salmon and L. Wendling. Arg based on arcs and segments to improve the symbol recognition by genetic algorithm. In W. Liu, J. Lladós, and J.M. Ogier, editors, *Graphics Recognition. Recent Advances and New Opportunities, Lecture Notes In Computer Science*, volume 5046, pages 80–90. Springer-Verlag Berlin, Heidelberg, 2008.

[SWW07] A. Sun, X. Wang, and H. Wei. A Camera Based Digit Location and Recognition System for Garment Tracking. In Koichi Kise and David S. Doermann, editors, *Proc. of the 2nd. International Workshop on Camera-Based Document Analysis and Recognition*, volume 94-99, Curitiba, Brazil, September 2007.

[Tan92] T.N. Tan. Texture feature extraction via visual cortical channel modelling. *Pattern Recognition, 1992. Vol.III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on*, pages 607–610, Aug-3 Sep 1992.

[Tan96] T.N. Tan. Written language recognition based on texture analysis. *Image Processing, 1996. Proceedings., International Conference on*, 1:185–188 vol.2, Sep 1996.

[TC88] C.-H. Teh and R.T. Chin. On image analysis by the methods of moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4):496–513, Jul 1988.

[TJ98] Mihran Tuceryan and Anil K. Jain. *The Handbook of Pattern Recognition and Computer Vision*, chapter Texture Analysis, pages 207–248. World Scientific Publishing, 2nd edition, 1998.

[TSM06] Fubito Toyama, Kenji Shoji, and Juichi Miyamichi. Symbol recognition of printed piano scores with touching symbols. In *ICPR*, volume 2, pages 480–483. IEEE Computer Society, 2006.

[TT89] Y.T. Tsay and W.H. Tsai. Model-guided attributed string matching by split-and-merge for shape recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 3(2):159–179, 1989.

[TTD06] K. Tombre, S. Tabbone, and P. Dosch. Musings on Symbol Recognition. *Lecture Notes in Computer Science*, 3926:23–34, 2006.

[TV06] O. Ramos Terrades and E. Valveny. A new use of the ridgelets transform for describing linear singularities in images. *Pattern Recogn. Lett.*, 27(6):587–596, 2006.

[TW02] S. Tabbone and L. Wendling. Technical symbols recognition using the two-dimensional radon transform. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 200–203, 2002.

[TWS06] S. Tabbone, L. Wendling, and J.P. Salmon. A new shape descriptor defined on the Radon transform. *Computer Vision and Image Understanding*, 102(1):42–51, 2006.

[US98] S. Uchida and H. Sakoe. A monotonic and continuous two-dimensional warping based on dynamic programming. In *Proceedings of 14th International Conference on Pattern Recognition*, volume 1, pages 521–524, 1998.

[VBB04] A. Vinciarelli, S. Bengio, and H. Bunke. Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 709–720, 2004.

[VGPS07] G. Vamvakas, B. Gatos, S. Petridis, and N. Stamatopoulos. An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition. In *Document Analysis and Recognition (ICDAR). Ninth International Conference on*, volume 2, pages 1073–1077, 2007.

[VGSP08] G. Vamvakas, B. Gatos, N. Stamatopoulos, and SJ Perantonis. A Complete Optical Character Recognition Methodology for Historical Documents. In *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on*, pages 525–532, 2008.

[VLOK00] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. Controlling On-Line Adaptation of a Prototype-Based Classifier for Handwritten Characters. *Proceedings of the 15th International Conference on Pattern Recognition*, 2:331–334, 2000.

[VLOK01] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters. *International Journal on Document Analysis and Recognition*, 3(3):150–159, 2001.

[VM00] E. Valveny and E. Marti. Hand-drawn symbol recognition in graphic documents using deformable template matching and a bayesian framework. *Proceedings of the 15th International Conference on Pattern Recognition*, 2:239–242, 2000.

[Vuo02] V. Vuori. *Adaptive Methods for On-Line Recognition of Isolated Handwritten Characters*. Published by the finnish academies of technology, Helsinki University of Technology, Acta Polytechnica Scandinavica, Mathematics and Computing Series, Helsinki University of Technologyechnica Scandinavica, Mathematics and Computing Series, 2002.

[WSR96] G. Wilfong, F. Sinden, and L. Ruedisueli. On-line recognition of handwritten symbols. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):935–940, 1996.

[WSV03] M. Wirotius, A. Seropian, and N. Vincent. Writer identification from gray level distribution. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 1168–1172, 2003.

[XCJW02] G. Xin, L. Cuiyun, P. Jihong, and X. Weixin. HMM based online hand-drawn graphic symbol recognition. In *Proceedings of the 6th*

International Conference on Signal Processing, volume 2, pages 1067–1070, 2002.

[Yan05] S. Yang. Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor. *IEEE transactions on pattern analysis and machine intelligence*, 27(2):278–281, 2005.

[YNGR07] R.B. Yadav, N.K. Nishchal, A.K. Gupta, and V.K. Rastogi. Retrieval and classification of shape-based objects using Fourier, generic Fourier, and wavelet-Fourier descriptors technique: A comparative study. *Optics and Lasers in engineering*, 45(6):695–708, 2007.

[YZL07] Y. Yu, W. Zhang, and W. Liu. A New Syntactic Approach to Graphic Symbol Recognition. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 1, pages 516–520, Brasil, September 2007. IEEE Computer Society Washington, DC, USA.

[ZA00] EN Zois and V. Anastassopoulos. Morphological waveform coding for writer identification. *Pattern Recognition*, 33(3):385–398, 2000.

[ZCB06] M. Zimmermann, J.C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. *IEEE transactions on pattern analysis and machine intelligence*, 28(5):818–821, 2006.

[ZL04] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.

[ZLZ06] W. Zhang, W.Y. Liu, and K. Zhang. Symbol recognition with kernel density matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2020–2024, December 2006.

[ZS03] B. Zhang and S.N. Srihari. Binary vector dissimilarity measures for handwriting identification. In *Proceedings of SPIE*, volume 5010, page 28, 2003.

[ZT06] D. Zuwala and S. Tabbone. A Method for Symbol Spotting in Graphical Documents. *LECTURE NOTES IN COMPUTER SCIENCE*, 3872:518–528, 2006.

[ZTW01] Yong Zhu, Tieniu Tan, and Yunhong Wang. Font recognition based on global texture analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1192–1200, Oct 2001.

Final Acknowledgment

This work has been partially supported by the Spanish projects TIN2006-15694-C02-02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018), and the catalan Fellowship 2007 BE-1 00086.