Analysis and Recognition of Music Scores 22

Alicia Fornés and Gemma Sánchez

Contents

Abstract

The analysis and recognition of music scores has attracted the interest of researchers for decades. Optical Music Recognition (OMR) is a classical research field of Document Image Analysis and Recognition (DIAR), whose aim is to extract information from music scores. Music scores contain both graphical and textual information, and for this reason, techniques are closely related to

A. Fornés (✉) • G. Sánchez
Computer Vision Center & Computer Science Department, Universitat Autònoma de Barcelona, Bellaterra, Spain
e-mail: afornes@cvc.uab.es; gemma@cvc.uab.es

D. Doermann, K. Tombre (eds.), *Handbook of Document Image Processing and Recognition*, DOI 10.1007/978-0-85729-859-1_24,

graphics recognition and text recognition. Since music scores use a particular diagrammatic notation that follow the rules of music theory, many approaches make use of context information to guide the recognition and solve ambiguities. This chapter overviews the main Optical Music Recognition (OMR) approaches. Firstly, the different methods are grouped according to the OMR stages, namely, staff removal, music symbol recognition, and syntactical analysis. Secondly, specific approaches for old and handwritten music scores are reviewed. Finally, online approaches and commercial systems are also commented.

Keywords

Graphics recognition • Optical music recognition • Staff removal • Symbol recognition

Introduction

Optical Music Recognition (OMR) is a classical area of interest of Document Image Analysis and Recognition (DIAR) that combines textual and graphical information. The recognition of music score images has been a very active research topic. OMR consists in the understanding of information from music scores (see an example in Fig. 22.1) and its conversion into a machine readable format. It allows a wide variety of applications [5] such as the edition of scores never edited, renewal of old scores, conversion of scores into braille, production of audio files and creation of collecting databases to perform musicological analysis.

Music Scores: Problem Definition

Music scores are a particular kind of graphical document, which include text and graphic elements. Graphical information appearing in music scores follows some 2D structural rules and corresponds to music information such as notes, staffs, and rests (silences). Textual information corresponds to lyrics and annotations such as title, author name, dynamic, and tempo markings, although their presence is not mandatory.

OMR has many similarities with Optical Character Recognition (OCR), because whereas OCR recognizes characters in text, OMR recognizes musical symbols in scores. It is nevertheless true that OMR belongs to graphics recognition because it requires the understanding of two-dimensional relationships, and music elements are two-dimensional shapes. The most common music symbols in a music score are notes, rests, accidentals, and clefs (see Fig. 22.2). Some terminologies used in music notation are the following:

- Staff: Five equidistant, horizontal lines where musical symbols are written down. 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.

Fig. 22.1 Example of a handwritten music score

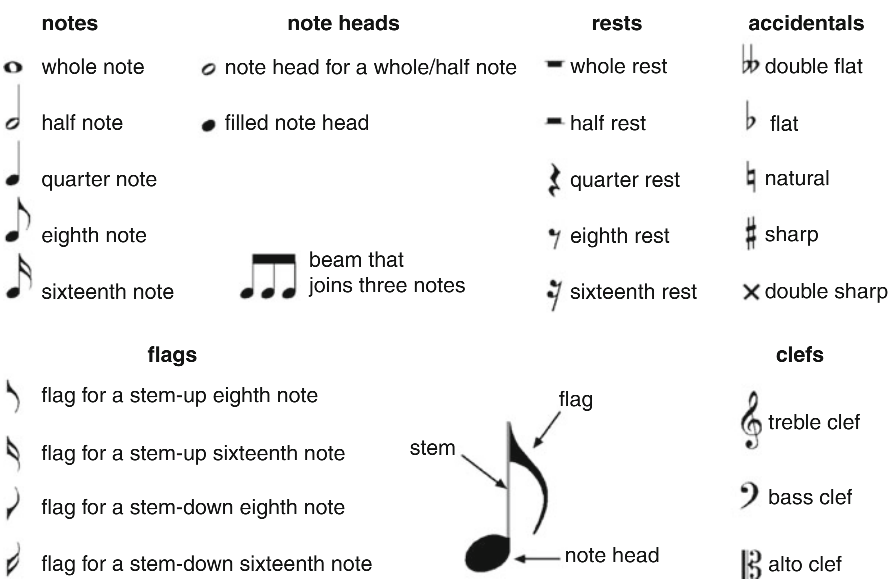

Fig. 22.2 Common elements of music notation

- Bar lines: Vertical lines which separate every bar unit or measure.
- Notes and rests (pauses): Notes are composed of head notes, beams, stems, flags, and accidentals.
- 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.

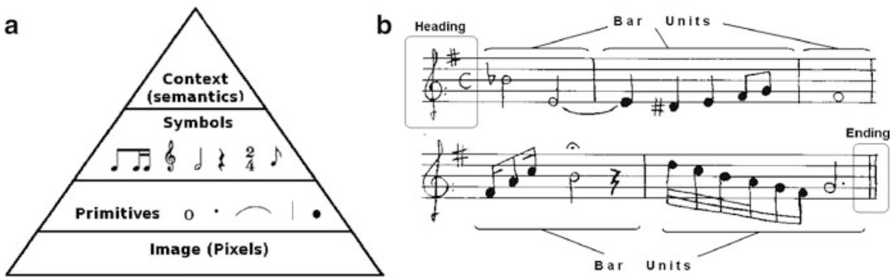

Fig. 22.3 OMR: (**a**) Levels of an OMR system, (**b**) structure of a music score

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. 22.3a). Context information helps to correct errors, and whereas dictionaries are commonly used in OCR, the formal music language theory is used in OMR.

A music score mainly consists in three blocks (see Fig. 22.3b): heading, bar units, and ending. Heading consists in the clef (alto, treble, 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.

History and Importance of OMR

From its beginning in 1966, Optical Music Recognition has been an active research field in the domain of document analysis. The first works started in the Massachusetts Institute of Technology by Pruslin in [46], followed by the DO-RE-Mi system proposed by Prerau in [45]. Already in these previous works, the basic stages and challenges for printed music scores were established. First, Pruslin [46] explained what an OMR system should give as output notes, and in what order they are played, their time values or durations, a volume, tempo, and interpretation. Later, Prerau [45] added the need to have a modular system, as the set of music symbols is very large and his work was using only a subset of them. These works started with a first step which scans and binarizes music scores, stressed how staff lines hindered the recognition of symbols by overlapping them, and finished by recognizing music symbols and testing the operation of their system. Just after these works were presented, a musician called Kassler [32] gave his opinion about them. Under his point of view, it was already clear that a computer could convert printed music in a computer readable form after recognizing the music symbols appearing in it. His worries were that these works only cope with a subset of all possible musical symbols and not the global notation. For that reason he appreciated the modular design of Prerau and its possibility to be increased with more symbols. Both works solved the problem for a subset of music symbols, but only Prerau's system was modular and extensible with more symbols. The problem with Pruslin's work was the preliminary step to remove horizontal and vertical lines in order to obtain isolated symbols. This process distorted so much some symbols that it was not possible to recognize them afterwards. In order to solve this problem, Prerau proposed a new process to detect symbols by using a split and merge process (see section "Staff Detection and Removal" for more details about these works).

During the decade of 1980s, several approaches appeared, basically focused on staff removal and the recognition of music symbols. Both Mahonney [38] and Roach and Tatem [53] proposed the recognition of music symbols by joining graphical primitives. Contrary, Clarke et al. [9] proposed the recognition of music symbols by analyzing the size and the bounding box of the symbols, very similar to the approach of Prerau. Mahoney also stressed the importance of the distinction between two kinds of music symbols: the ones describing "what" is to be played (e.g., notes, accidentals) and the ones describing "how" music has to be played (e.g., dynamics, tempo markings). Most of previous works were only focused on what is to be played as notes, clefs, and so on. In this decade, it is remarkable the work of Andronico and Ciampa [1], who proposed the use of grammars to guide the recognition. In this way, syntactic information could be used to solve ambiguities. Finally, it must be commented that in 1989, the robot Wabot-2 [39] was created. It was an almost real-time keyboard-playing robot able to read simple music scores and play the organ.

Since first works in 1970s and 1980s, the interest of OMR has grown considerably. Most works through 1990s focused on removing staff lines and recognizing music symbols with the help of grammars and syntactical rules [10, 19, 42], following the idea of Andronico in 1982. Thus, systems could take advantage of music notation theory in order to add syntactical information that guides the recognition and solves ambiguities and inconsistencies. Also during the first decade of 2000, many approaches appeared that dealt with the recognition of polyphonic and complex music scores, such as the work of Kato and Inokuchi [33], who could handle complex music notation such as two voices per staff, chords, shared noteheads, slurs, and pedal markings. Nowadays, the OMR systems of printed scores include effective algorithms (see Gamera and Aruspix systems [49]) to recognize and interpret the resulting 2D arrangement of symbols and precise formalisms for representing the results of interpretation. However, it is interesting to remark that the staff removal problem is still attracting the interest of researchers in OMR, with the proposal of many algorithms for removing staff lines [12, 26], especially for old and handwritten music scores [14, 16, 20, 56].

Contrary to printed scores, the recognition of handwritten music scores is still considered to be an open problem. To the best of our knowledge, the first attempt was performed by Ng in [43], who discussed the adaptation of OMR for printed music scores to handwritten ones. The recognition system for printed music

Table 22.1 History and evolution of the optical music recognition approaches and the main type of research: *STF* staff removal, *SYM* music symbol recognition, *HW* handwritten music scores, *GR* grammars and rules, *ON* online OMR

Year	Work	Type	Year	Work	Type
1966	Pruslin [46]	STF, SYM	2001	Ng [43]	HW
1970	Prerau [45]	STF, SYM	2003	Pinto et al. [44]	SYM, HW
1982	Mahoney [38]	STF, SYM	2004	Fujinaga [26]	STF
1982	Andronico and Ciampa [1]	GR	2005	Rossant and Bloch [54]	SYM
1988	Clarke et al. [9]	STF, SYM	2005	Macé et al. [37]	ON
1988	Roach and Tatem [53]	STF, SYM	2006	Fornés et al. [20]	STF, HW
1989	Matsushima et al. [39]	GR	2006	Toyama et al. [57]	SYM
1991	Kato and Inokuchi [33]	STF, GR	2006	Homenda and Luckner [31]	SYM
1992	Carter and Bacon [8]	STF, SYM	2006	Pugin [47]	SYM
1993	Randriamahefa et al. [50]	STF, SYM	2006	Miyao and Maruyama [40]	ON
1993	Leplumey et al. [35]	STF	2008	Dalitz et al. [12]	STF
1993	Modayur et al. [42]	GR	2008	Pugin et al. [49]	SYM
1993	Fahmy and Blostein [19]	GR	2009	Cardoso et al. [14]	STF
1995	Baumann [4]	GR	2009	Escalera et al. [17]	SYM
1995	Miyao and Nakano [41]	SYM	2010	Dutta et al. [16]	STF
1995	Couasnon and Rétif [10]	GR	2010	Fornés et al. [21]	SYM, HW
1997	BainBridge and Carter [3]	STF,SYM	2010	Rebelo et al. [51]	SYM, HW
1999	Stückelberg and Doermann [55]	SYM	2011	Escalera et al. [18]	SYM, HW

was based in the detection of graphical primitives in order to regroup them and form music symbols. However, he concluded that this technique was not robust enough for dealing with music symbols because of the high variability in the handwriting styles. Since then, some other approaches for recognizing handwritten music symbols have appeared [21, 44, 51], although most of these systems are only able to recognize a small set of music symbols and not the whole music score.

Some interesting surveys of classical OMR can be found in [5, 30, 52], where several methods to segment and recognize symbols are reviewed. The evolution of OMR is shown in Table 22.1.

Summary of the State of the Art

The research on Optical Music Recognition could be classified considering the different OMR stages, namely, preprocessing, staff removal, symbol recognition, and validation. Thus, in the next section some of the main works that have been proposed for each one of the these stages are reviewed.

However, and since the above approaches must be adapted to the kind of document (e.g., typewritten, handwritten, old, online), section "State of the Art in Different Domains" is devoted to the specific works for dealing with old, handwritten, and online music documents.

State of the Art in the Different Stages of OMR

Introduction

The main stages of an Optical Music Recognition (OMR) system are the following: preprocessing, staff removal, symbol recognition, and validation (semantics). It is important to remark that in many cases, the separation between the stages is difficult because some OMR approaches join some of these stages together. For example, the authors of [10] and [33] propose a validation stage that integrates the symbol recognition module. Also, there are few approaches [39] and [47] that do not have any staff removal step.

Preprocessing

The preprocessing step usually consists in binarization, skew correction, and noise removal. Since the preprocessing techniques for music scores are usually generic document image analysis techniques, the reader is referred to ▸Chap. 4 (Imaging Techniques in Document Analysis Processes) and [52] for more information.

Staff Detection and Removal

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. For this purpose, staff removal algorithms must cope with gaps, crossing symbols, overlapping symbols (see Fig. 22.4a), and in case of staff lines written by hand (see Fig. 22.4b), also with distorted lines.

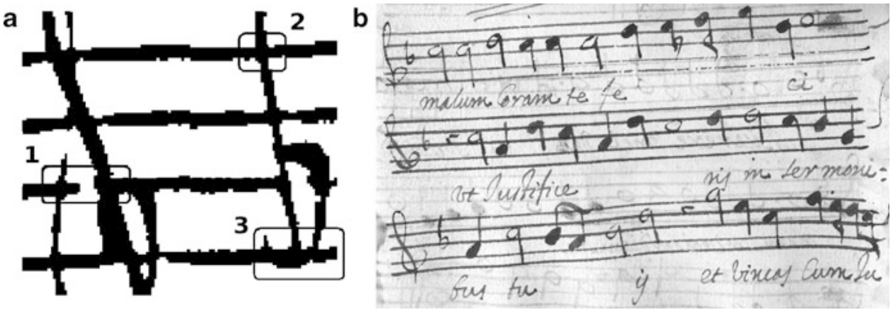

Fig. 22.4 Typical problems related to the removal of staff lines: *1*. gap, *2*. crossing symbol, *3*. overlapping symbol. (**b**) Old Spanish music score from the seventeenth century, where the staff lines are written by hand. Notice that the staff lines are not equidistant nor parallel and lines have different widths

Typically, staff removal methods are based on projections and run-length analysis, contour-line tracking, or graphs. Next, some of the most common approaches are described.

Projections and Run Lenghts

The first group of methods are based on projections and/or run-length analysis, which are very fast to compute. Methods based on projections require horizontal staff lines. Thus, after deskewing the document, projections (Y projection) are used to recognize staff lines [44]. A defined threshold is usually used to select projections strong enough to be candidate staff lines. These candidates are searched to find groups of five equally spaced lines.

The first staff removal approach, the one proposed by Pruslin [46] in 1966, consisted in eliminating all thin horizontal and vertical lines, including many bare staff-line sections and stems. This results in an image of isolated symbols, such as noteheads and beams, which are then recognized using contour-tracking methods. As expected, this preprocessing step erases or distorts most music symbols.

The method proposed by Fujinaga in [26] detects staves by horizontal projections, but deskews each staff separately. Afterwards, black vertical runs larger than a threshold are removed and consider all remaining connected components with a considering width. Cui et al. [11] propose a method based on the detection and selection of region of interests (ROI). After rotation correction, the staff-line width is estimated using horizontal projection and the staff lines are removed correspondingly. Randriamahefa et al. [50] propose a more complex projections method. It consists in a vertical projection, projection filtering, local minima region finding (the peaks will probably correspond to the staff lines), horizontal projection of each local minima regions (to ensure that those peaks are certainly lines), and linking different peaks between them in order to build up the staff (a staff implies five staff lines). These techniques are usually very robust because even if these lines are bowed, skewed, or fragmented, the staff lines are always found. In most cases, after the detection of the staff lines, the next step consists in estimating the width of the staff line and then erases the line segments that are smaller than a threshold (proportional to the estimated thickness).

Other methods also perform a run-length analysis. For example, Kato and Inokuchi [33] detect and extract staff lines using histograms, run lengths, and projections. Afterwards, the staff is analyzed (tracking from the left), eliminating short horizontal runs whose width is under a certain threshold. In the staff removal method of Dutta et al. [16], a staff-line segment is considered as a horizontal linkage of vertical black runs with uniform height. Then, the neighboring properties of a staff-line segment are used to validate it as a true segment. Similarly, in the approach of Clarke et al. [9], 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 the analysis 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 staves. Complete staff separation is not always achievable, because parts of symbols belonging to the staff above or the staff below may be included.

The method proposed by Su et al. [56] first estimates the staff height and space by using the histogram of vertical run length. Afterwards, an initial staff line is modeled, which predicts the line direction and fits an approximate staff-line curve for each image. The fitted staff-line curve can be used to identify the actual location of staff lines on the image, where pixels belonging to staff lines are removed.

Candidates Assemblage and Contour Tracking

These groups of methods usually detect candidate staff-line segments (usually from the skeleton of the image), which are afterwards analyzed using contour tracking.

Prerau [45] 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 has 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 of this technique is that symbols which merge with staff lines do not always have horizontal overlap, so with this method, symbols would keep disconnected when they should be connected.

In the work of Roach and Tatem [53], 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 systems proposed by Mahoney [38], Fornés et al. [20] and Dalitz et al. [12] create staff-line candidates from the image. Afterwards, these candidates are analyzed, and some rules are applied (allowable thicknesses, orientations, lengths, and gap lengths) in order to discard or join them. 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.

In the prototype for handwritten scores presented by Ng in [43], 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 a part of a staff-line using basic notational syntax and an estimated staff line height.

Graph Path Search

The last group of staff removal methods propose the creation of a graph.

Carter and Bacon (see [8]) 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, most 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° and copes with slight bowing of staff lines and with local variations in staff-line thickness.

Leplumey et al. [35] present a method based on a prediction-and-check technique to extract staves, 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 are 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.

Cardoso et al. [14] propose a graph-theoretic framework where the staff line is the result of a global optimization problem. The staff-line algorithm uses the image as a graph, where the staff lines result as connected paths between the two lateral margins of the image. A staff line can be considered a connected path from the left side to the right side of the music score. The main cycle of the methodology consists in successively finding the stable paths between the left and right margins, adding the paths found to the list of staff lines, and erasing them from the image. To stop the iterative staff-line search, a sequence of rules is used to validate the stable paths found.

No Staff Removal

Finally, there is a small group of approaches that do not remove staff lines, because they need a real-time OMR system [39] or because they would like to avoid the

Table 22.2 Staff removal comparison of the main group of methods. It shows their performance when dealing with overlapping symbols with staff lines, grouping five staff lines into a staff, and the (estimated) speed of the algorithm. Note that − means poor performance and ++ means very good

Method	Works	Overlapping symbols-staff	Staff-lines grouping	Speed
Projections, histograms, and run lengths	[9, 34, 46], [26, 44, 50], [11, 16, 33, 56]	−	++	++
Candidates assemblage and contour tracking	[20, 38, 53], [12, 43, 45]	+	+	+
Graph path search	[8, 14, 35]	++	+	−

segmentation problems that the staff removal may cause [47]. The Wabot-2 robot of Matsushima et al. [39] 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 staves). 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 notehead size. Afterwards, the image of each staff is normalized according to these parameters.

Summary

A comparison of the different kind of staff removal algorithms is shown in Table 22.2. Systems based on projections and run lengths are fast to compute, but since they usually split symbols that were touching the staff lines, the posterior recognition stage must deal with broken symbols. Contrary the performance of contour-tracking and graph-based techniques is usually better, especially concerning overlapping symbols with staff lines.

Concerning ground truthing for testing the staff removal algorithms, Dalitz et al. [12] have created distorted images from printed music scores. In order to test these algorithms in handwritten music scores, the reader is referred to the CVCMUSCIMA database [24]. Finally, it must be said that a comparative study of four staff removal algorithms can be found in [12]. In addition, two staff removal competitions had been recently organized in ICDAR (International Conference on Document Analysis and Recognition) [22] and GREC (International Workshop on Graphics Recognition) [23]. In both cases, the conclusion is that there is no algorithm that performs best in all cases (depending on the distortions to deal with, one algorithm performs better than others), showing that there is still room for improvements.

Fig. 22.5 Composition of musical symbols. (**a**) Two equivalent music symbols. (**b**) Combinations of 8th and 16th notes

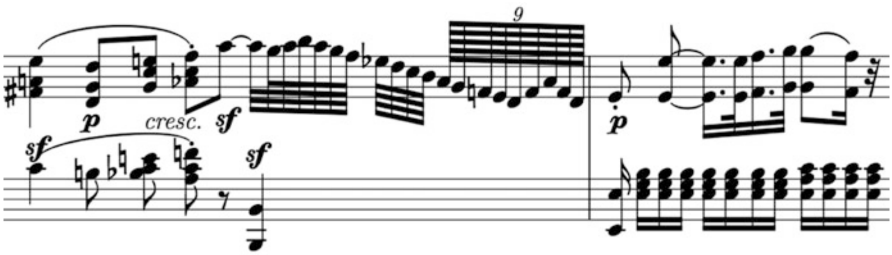

Fig. 22.6 Compound notes extracted from Sonata "Pathetique" from Beethoven

Music Symbol Extraction and Classification

The classification of music symbols could be treated as a symbol recognition problem, and therefore, some symbol recognition techniques have been applied. However, simple music symbols are usually combined following certain rules for creating complex compound music symbols (see Fig. 22.5), and for this reason, many techniques based on the analysis of graphical primitives and their combinations have been proposed. Thus, they can cope with the almost infinite variability of compound symbols (see Fig. 22.6).

The main techniques have been classified into different groups, which are described next.

Template Matching

Template matching means comparing the different region object candidates with the templates that represent the different music symbols. Since it is computationally expensive and sensitive to variations in shape, it has been applied in few cases. For example, Pruslin [46] uses contour tracking to describe connected binary image regions which remain after deleting horizontal and vertical lines. Classification depends both on stroke properties and on inter-stroke measurements (a method for template matching using contour strokes is developed). Toyama et al. [57] present a symbol recognition method for printed piano music scores with touching symbols. 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.

The Wabot-2 robot [39] recognizes the musical symbols (with staff lines) according to a two-level hierarchy: the upper level, corresponding to the recognition of staff lines, noteheads, and bar lines, is implemented in hardware and the lower level in software. The search is performed using hardware-implemented template matching.

Simple Operations

This kind of methods bases the recognition of music symbols with simple operations, such as the analysis of bounding boxes, projections, or morphological operations.

Prerau [45] and Clarke et al. [9] use the relative symbol size for an initial classification. In [45], the dimensions of the bounding box are used to look up a list of possible matches (there is a precomputed 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. Contrary, Clarke et al. [9] do not use any lookup table. Instead, pixels in few particular rows and columns of the symbol image are examined.

Lee and Choi [34] use projection methods to recognize staff lines, bar lines, notes, and rests. After detecting the staff lines, 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, where they are compared with the stored projections for known symbols. The main disadvantage of this method is that it is rotation sensitive.

In the works of Bainbridge and Carter [3] and Carter and Bacon [8], objects are classified according to the bounding-box size and according to the number and organization of their constituent sections.

The system proposed by Modayur et al. [42] is composed of two modules: the low-level vision module uses morphological algorithms for symbol 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.

Joining Graphical Primitives

These systems extract graphical primitives (e.g., noteheads, stems beams, and flags) and combine them in order to form music symbols, such as notes, chords, and beamed note sequences.

Mahoney [38] does not use context information for the recognition of primitives, but it is used to infer musical symbols from the relationships between the different kind 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.

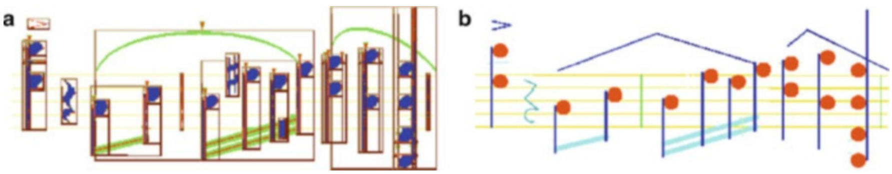

Fig. 22.7 Classification of musical symbols performed in [43]. (**a**) Oversegmentation of symbols, (**b**) recognized primitives

In the system proposed by Roach and Tatem [53], knowledge about music notation is represented in a rule-based system, which is applied starting with the earliest steps of symbol segmentation and recognition. 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. Since notehead detection is extremely difficult in handwritten music scores, a general-purpose blob detector does not work. Thus, in [20] and [53], noteheads are searched in constrained locations: first, verticals lines are located, and 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 notehead. Also, Kato and Inokuchi [33] describe a primitive extraction stage which analyzes the size of lines, blobs, and circles in order to detect noteheads and stems candidates that will be analyzed in the next stages. Similarly, Ng exposes in [43] a prototype for printed scores, where a combination of edges, curvature, and variations in relative thickness and stroke direction is applied to oversegment symbols into lower-level graphical primitives (lines, curves, and ellipses). Afterwards, these primitives will be joined to form music symbols. An example of the detected primitives is shown in Fig. 22.7.

A more complex approach is proposed by Miyao and Nakano [41] for the extraction of heads and stems in printed piano scores. After extracting all region candidates to be 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 than the shapes of heads. Afterwards, the networks are used to identify the head candidates. Finally, the stem candidates touching the detected heads are extracted as true stems.

Statistical Approaches

The authors of [51] perform a comparative study of four classification methods for musical primitives: Neural Networks (NN), Support Vector Machines (SVM), k-Nearest Neighbor (k-NN), and Hidden Markov Models (HMM). In the first three methods, the input features are the raw pixels of the image (resized at 20×20), whereas for HMM, high-level features (see [47]) are extracted from a sliding window. Results show that the best classification method is SVM, closely followed by k-NN.

Symbol Descriptors

This kind of methods uses symbol descriptors (see ▸Chap. 16 (An Overview of Symbol Recognition)) for the recognition of music symbols. For example, Homenda and Luckner [31] 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) and use Principal Component Analysis for improving the final recognition rate. In [17] and [18], Escalera et al. describe two symbol recognition methods that can be applied to hand-drawn music symbols. In both methods, the descriptor defines a probability density function of the shape of the symbol using a rectangular [17] or circular grid [18]; and Adaboost learns the discriminative features that better split symbol classes. Also, Fornés et al. describe a rotation symbol recognition method for recognizing hand-drawn music symbols [21]. The method extracts column sequences of feature vectors from different orientations. Afterwards, the Dynamic Time Warping algorithm is used for finding the best matching between the two symbols to be compared.

Structural and Syntactical Approaches

The last set of methods (grammars, graphs, fuzzy modeling, etc.) uses context information in order to solve ambiguities. Most of them integrate the recognition of symbols with the validation (syntactic and semantic) stage.

Stückelberg and Doermann [55] propose a probabilistic framework for the recognition of printed scores. The modeling structure is similar to a stochastic attribute grammar, where local parameters are estimated using Hidden Markov Models (HMM). It also uses the Hough Transform and a low-band filter to locate lines and noteheads of note groups.

Randriamahefa et al. [50] propose an attributed graph, which is constructed from the skeleton of the image: the graph nodes correspond to 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.

Rossant and Bloch [54] propose a system based on a fuzzy modeling of symbol classes and music writing rules. 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.

Table 22.3 Recognition of music symbols. Comparison of the main kind of methods, showing their robustness to noise, the estimated speed, and the robustness to different types, fonts, or handwriting styles. Note that — means poor performance and ++ means very good

Method	Works	Appearance robustness	Noise robustness	Speed
Template matching	[39, 46, 57]	−−	−−	−
Bounding box, projections morpholog. operations	[3, 9, 45], [8, 34, 42]	−−	−−	++
Joining graphical primitives	[38, 41, 53], [20, 33]	+	+	+
Statistical	[47, 51]	+	++	−
Symbol descriptors	[17, 31], [18, 21]	++	++	+
Syntactic and structural	[50, 54, 55]	++	++	−

Summary

Table 22.3 shows a comparison of the different approaches for the extraction and classification of music symbols. It compares the estimated speed of the system, the robustness to noise, and the sensitivity to the variability of the appearance (e.g., font of the publisher, handwriting variabilities).

Since template matching is computationally very expensive, it can be effectively used when two constraints are satisfied: first, no real-time algorithm is needed (or there is dedicated hardware for this), and second, all the music symbols belonging to the same class look very similar. In contrast, methods based on simple operations are very fast to compute. However, both kinds of methods (template matching and simple operations) are very sensitive to noise. Consequently, the performance depends on the font of the publisher and will not work in handwritten scores.

Methods based on joining graphical primitives are more robust to the variability of symbols' shape, but they cannot deal with symbols that cannot be discomposed into primitives (e.g., music clefs). The recognition of these symbols can be solved with the application of symbol descriptors. However, the best results are usually obtained from methods that make use of context information (rules) such as syntactic and structural approaches (although this usually implies decreasing the speed of the system).

Syntactical Analysis and Validation

Rules on music notation make 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 [1, 45] use two different grammar levels: lower-level grammars for music symbols (with terminal geometric figures such as dots, circles, and lines and adjacency operations such as above, below, right of, etc.) and high-level grammars for music sentences (larger units with measures containing music symbols).

Grammars

The first group of methods use a grammar to guide the recognition. In [39] 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 the following: a fat double bar appears only at the end of each part, a clef always appears right at the beginning of each staff, the number of beats in each measure should match the time signature, etc.

In [10], 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.

Rules or Graph Grammars

The second group of methods is based on rules or constraints. Modayur et al. [42] propose a high-level reasoning module. 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 component analysis and a reasoning module (which 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.

Kato and Inokuchi [33] describe a sophisticated top-down architecture. 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, symbol synthesis, symbol recognition, and semantic analysis) have recognition and verification units. The primitive extraction module contains units for recognizing stems, beams, and noteheads. Hypothesized primitives are removed from the pixel image. Unacceptable hypotheses are rejected at higher layers and 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.

Baumann proposed in [4] the DO-RE-MI-DI++ system, which allows multiple score layouts and polyphonic input. This system allows the definition of syntactic and semantic knowledge by specifying graph-grammar rules. Similarly, Fahmy and Blostein [19] present a graph grammar for recognizing musical notation, 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 does not need to be specified.

State of the Art in Different Domains

Introduction

The difficulties found in the recognition of music scores strongly depend on the kind of music scores. For example, the recognition of old documents has to cope with paper degradation, whereas handwritten scores mean dealing with the variability in the handwriting style. In case of old handwritten music scores (see an example in Fig. 22.8), the systems must cope with both problems.

Next, the main approaches for the recognition of old scores, handwritten scores, and also online handwritten music scores are reviewed.

Old Handwritten Music Scores

There are few works concerning the recognition of old music scores. They are briefly described next.

Bainbridge, Carter et al. propose a system for recognizing early music [2, 3, 8]. The system described in [7] is used to recognize old music scores, corresponding to madrigals (see Fig. 22.9) notated in *White Mensual Notation*. Symbols are correctly segmented and an early classification stage has been implemented.

Fig. 22.8 Old handwritten music score

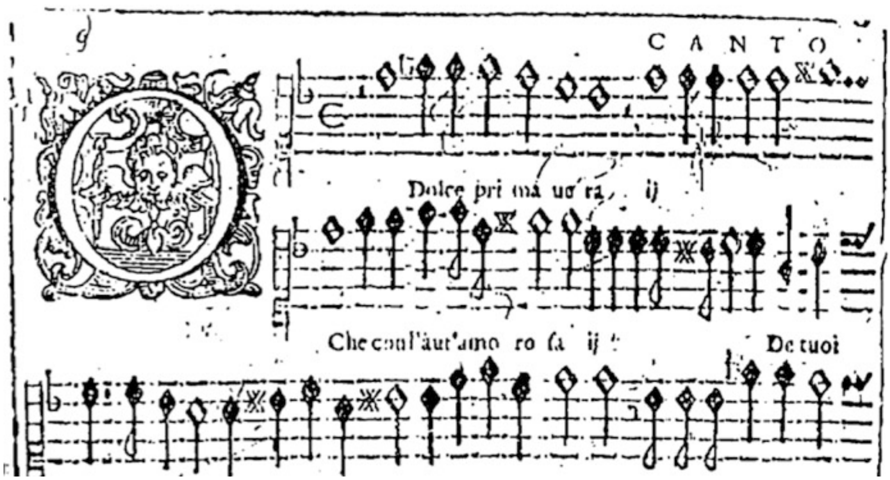

Fig. 22.9 Example of an old madrigal, extracted from [7]

The approach first removes the staff lines using Line Adjacency Graph (LAG), which has been explained in the staff removal subsection. Thus, the resulting image contains music symbols or connected components that belong to music symbols. In the classification stage, these elements are classified according to the bounding-box size and the number and organization of their constituent sections. The author states that if there are overlapping or superimposed symbols, another algorithm will be required.

Pinto et al. [44] proposes an OMR method to recognize ancient musical scores. This system copes with specific notation of ancient documents. After the preprocessing stage, the segmentation module divides the music sheet in staff lines, bars, and musical symbols. The staff lines are removed using horizontal projections. 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).

Fornés et al. [20] propose a method for the recognition of graphical primitives in old handwritten music scores. After removing the staff lines using contour tracking, the system detects vertical lines, bar lines, and filled head notes using median filters and morphological operations. The system also recognizes music clefs using Zernike moments. The authors suggest that a grammar should be incorporated in order to cope with the important amount of false positives obtained.

An approach for OMR in printed scores from the 16th to 17th is presented by Pugin [47]. The system consists in a segmentation-free approach based on Hidden Markov Models (HMM). They do 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. For feature extraction, they use a sliding window as in speech recognition, extracting the following six 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 the 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.

Finally, an interesting comparison of two OMR approaches (Gamut and Aruspix) applied to ancient scores can be found in [49], where the authors demonstrate that paper degradation affects to the final performance. They also perform an evaluation of binarization techniques for OMR in [48].

Modern Handwritten Music Scores

Ng exposes in [43] a prototype for printed scores, followed by a prototype for handwritten ones, discussing the limitations of the first one for handwritten scores processing. From the skeleton of the image, staff lines are detected and removed. Then, a combination of edges, curvature, and variations in relative thickness and stroke direction is used to perform further subsegmentation and segregate the writings into lower-level graphical primitives (lines, curves, and ellipses). Afterwards, primitives are classified using a k-NN 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 subsegmented 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 regroupings. Unluckily, no recognition rates are shown in the recognition of handwritten scores.

Rebelo et al. [51] tested the performance of four classification methods: Neural Networks (NN), Support Vector Machines (SVM), k-Nearest Neighbor (k-NN), and Hidden Markov Models (HMM) on 50 handwritten music scores. These methods have been described in the previous section. Results show that SVM and k-NN obtain the best results.

Fig. 22.10 The online system proposed by Miyao and Maruyama [40]

OnLine Music Scores

In the last decade, several online music recognition systems have appeared, although the amount of music symbols that can be recognized is still limited. George [27] presents a system based on Neural Networks, which resizes the input symbol, divides the symbol into 25 regions, and computes the density in each one. Afterwards, a multilayer perceptron is used for classification. For evaluating the system, the author has created a database or 4,188 music symbols, containing 20 different music symbols written by 25 people. The system obtains a recognition rate of about 80 %, using the 50 % of the symbols for training and the remaining 50 % for testing.

Macé et al. [37] propose an online system with three main components: a framework pen-based interaction, a formalism to describe the composition of documents, and a handwritten stroke parser. The formalism is based on context-free grammars, with chronological information, spatial structure, and composition rules. The handwritten stroke parser drives the different recognizers (e.g., alteration, stem, head) with empirical heuristics or Neural Networks; however, the authors do not provide detailed information about the recognizers. The framework allows the user to validate or reject an answer from the parser. The system can recognize whole, half, and quarter notes, accidentals, clefs, rests, and bar lines. Unluckily, the authors do not provide any recognition rates.

Miyao and Maruyama [40] present an online symbol recognition system which combines 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 Freeman 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 8×8 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 predefined table. To allow a stroke distortion, some music symbols have several possibility combinations of strokes. The proposed method reaches a recognition rate of 98 %. An example of the output of their system is shown in Fig. 22.10.

Conclusion

In this chapter, the main works in Optical Music Recognition have been reviewed. Firstly, the music notation and the history of the music recognition research field have been overviewed. Then, the main approaches in each stage of an Optical Music Recognition system have been described. The OMR stages consist of preprocessing, staff removal, symbol recognition, and validation. Finally, the main works concerning the recognition of old music scores as well as handwritten (off-line and online) music scores have been reviewed.

The research could be considered in a mature state concerning the recognition of printed scores, including staff removal, primitives, music symbol recognition, and, also, the validation of the music score.

Open problems are still the recognition of old scores, handwritten music scores, as well as complex music scores (scores with many voices, chords, etc.). In fact, current approaches can effectively deal with simple symbols, but the recognition of complex symbols is still a difficult task. Similarly, online approaches are still not able to effectively recognize complex symbols, and this is probably one of the main reasons why composers prefer to work on paper document yet. In addition, the current performance of the OMR systems is not as good as desired, and therefore they are still not applicable for productive use.

For these reasons, promising approaches are the ones that take into account the context information, the structure of the music score, and the music notation. In this sense, grammars have shown to be a good choice for guiding the recognition, discarding false detections, and helping with ambiguities. In addition, learning-based methods (such as Hidden Markov Models and Neural Networks) are promising research directions for coping with the variability of handwritten music scores.

Consolidated Software

There are several commercial OMR systems, such as:

- SharpEye: http://www.visiv.co.uk
- PhotoScore: http://www.neuratron.com/photoscore.htm
- SmartScore: http://www.musitek.com
- Audiveris (open source): http://audiveris.kenai.com

In most systems, the recognition rates significantly decrease when the document is noisy, degraded, or with low resolution. As far as the authors know, PhotoScore is the only one that can deal with handwritten music scores.

Some OMR systems are also available in the academic field. The Gamera framework [15] has been frequently used in OMR applications [12–14,16], although it has been designed for building document analysis applications. The authors of [49] compare two systems: Gamut (a Gamera application) and Aruspix. Both have been applied to ancient scores. The authors conclude that although Aruspix HMM models outperform the Gamut k-NN classifiers, experiments show that paper degradation affects to the performance of both systems.

Cross-References

►An Overview of Symbol Recognition
►Analysis and Interpretation of Graphical Documents
►Analysis of the Logical Layout of Documents
►Continuous Handwritten Script Recognition
►Graphics Recognition Techniques
►Handprinted Character and Word Recognition
►Online Handwriting Recognition
►Text Segmentation for Document Recognition

References

1. Andronico A, Ciampa A (1982) On automatic pattern recognition and acquisition of printed music. In: Proceedings of the international computer music conference, Venice, pp 245–278
2. Bainbridge D, Bell T (1996) An extensible optical music recognition system. In: Proceedings of the nineteenth Australasian computer science conference, Melbourne, pp 308–317
3. Bainbridge D, Carter N (1997) Automatic reading of music notation. In: Bunke H, Wang PSP (eds) Handbook of character recognition and document image analysis, Chapter 22. World Scientific, Singapore, pp 583–603
4. Baumann S (1995) A simplified attributed graph grammar for high-level music recognition. In: International conference on document analysis and recognition, Montreal, vol 2. IEEE Computer Society, Los Alamitos, pp 1080–1983
5. Blostein D, Baird HS (1992) A critical survey of music image analysis. In: Baird HS, Bunke H, Yamamoto K (eds) Structured document image analysis. Springer, New York, pp 405–434
6. Bruder I, Ignatova T, Milewski L (2004) 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), Tucson, AZ, USA, pp 397–397
7. Carter NP (1992) Segmentation and preliminary recognition of madrigals notated in white mensural notation. Mach Vis Appl 5(3):223–229
8. Carter NP, Bacon RA (1992) Automatic recognition of printed music. In: Baird H, Bunke H, Yamamoto K (eds) Structured document image analysis. Springer, Berlin Heidelberg, pp 456–465
9. Clarke A, Brown B, Thorne MP (1988) Inexpensive optical character recognition of music notation: a new alternative for publishers. In: Computers in music research conference, Bailrigg, Lancaster, pp 84–87

10. Coüasnon B, Rétif B (1995) Using a grammar for a reliable full score recognition system. In: International computer music conference, Banff, Canada, pp 187–194
11. Cui J, He H, Wang Y (2010) An adaptive staff line removal in music score images. In: IEEE 10th international conference on signal processing (ICSP), Beijing. IEEE Computer Society, pp 964–967
12. Dalitz C, Droettboom M, Pranzas B, Fujinaga I (2008) A comparative study of staff removal algorithms. IEEE Trans Pattern Anal Mach Intell 30(5):753–766
13. Dalitz C, Michalakis GK, Pranzas C (2008) Optical recognition of psaltic byzantine chant notation. Int J Doc Anal Recognit 11(3):143–158
14. dos Santos Cardoso J, Capela A, Rebelo A, Guedes C, Pinto da Costa J (2009) Staff detection with stable paths. IEEE Trans Pattern Anal Mach Intell 31(6):1134–1139
15. Droettboom M, MacMillan K, Fujinaga I (2003) The gamera framework for building custom recognition systems. In: Proceedings of the symposium on document image understanding technologies, Greenbelt, Maryland (USA), pp 7–11. Citeseer
16. Dutta A, Pal U, Fornés A, Lladós J (2010) An efficient staff removal approach from printed musical documents. In: International conference on pattern recognition, Istanbul. IEEE Computer Society, Istambul, Turkey, pp 1965–1968
17. Escalera S, Fornés A, Pujol O, Radeva P, Sánchez G, Lladós J (2009) Blurred shape model for binary and grey-level symbol recognition. Pattern Recognit Lett 30(15):1424–1433
18. Escalera S, Fornés A, Pujol O, Lladós J, Radeva P (2011) Circular blurred shape model for multiclass symbol recognition. IEEE Trans Syst Man Cybern B Cybern 41(2):497–506
19. Fahmy H, Blostein D (1993) A graph grammar programming style for recognition of music notation. Machine Vis Appl 6:83–99. Springer
20. Fornés A, Lladós J, Sánchez G (2006) Primitive segmentation in old handwritten music scores. In: Liu W, Lladós J (eds) Graphics recognition: ten years review and future perspectives. Volume 3926 of lecture notes in computer science, pp 279–290. Springer, Berlin Heidelberg
21. Fornés A, Lladós J, Sánchez G, Karatzas D (2010) Rotation invariant hand drawn symbol recognition based on a dynamic time warping model. Int J Doc Anal Recognit 13(3): 229–241
22. Fornés A, Dutta A, Gordo A, Llados J (2011) The icdar 2011 music scores competition: staff removal and writer identification. In: International conference on document analysis and recognition (ICDAR), Beijin, China. IEEE Computer Society, pp 1511–1515
23. Fornés A, Dutta A, Gordo A, Lladós J (2012) The 2012 music scores competitions: staff removal and writer identification. In: Kwon Y-B, Ogier J-M (eds) Graphics recognition. New trends and challenges. Lecture notes in computer science, vol 7423. Springer, Berlin/Heidelberg
24. Fornés A, Dutta A, Gordo A, Lladós J (2012) Cvc-muscima: a ground truth of handwritten music score images for writer identification and staff removal. Int J Doc Anal Recognit 15(3):243–251
25. Fornés A, Lladós J, Sanchez G, Bunke H (2012) Writer identification in old handwritten music scores. In: Papaodysseus C (ed) Pattern recognition and signal processing in archaeometry: mathematical and computational solutions for archaeology. IGI Global, Hershey, Pennsylvania, USA, pp 27–63
26. Fujinaga I (2004) Staff detection and removal. In: George S (ed) Visual perception of music notation. Idea Group, IRM Press, Hershey PA, USA, pp 1–39
27. George S (2003) Online pen-based recognition of music notation with artificial neural networks. Comput Music J 27(2):70–79
28. George S (2005) Visual perception of music notation: on-line and off-line recognition. IRM Press, Hershey PA, USA
29. Gordo A, Fornes A, Valveny E (2013) Writer identification in handwritten musical scores with bags of notes. Pattern Recognit 46:1337–1345. doi:http://dx.doi.org/10.1016/j.patcog.2012.10.013
30. Homenda W (2005) Optical music recognition: the case study of pattern recognition. In: Kurzynski M, Puchala E, Wozniak M, Zolnierek A (eds) CORES, advances in soft computing, Rydzyna Castle, Poland. Editorial: Springer, Berlin Heidelberg, vol 30. Springer, pp 835–842

31. Homenda W, Luckner M (2006) Automatic knowledge acquisition: recognizing music notation with methods of centroids and classifications trees. In: International joint conference on neural networks, Vancouver, Canada. IEEE Computer Society, pp 3382–3388
32. Kassler M (1972) Optical character recognition of printed music: a review of two disertations. Perspect New Music 11(2):250–254
33. Kato H, Inokuchi S (1991) A recognition system for printed piano music using musical knowledge and constraints. In: Baird HS, Bunke H, Yamamoto K (eds) Structured document image analysis. Springer, Berlin Heidelberg, pp 435–455
34. Lee MW, Choi JS (1985) The recognition of printed music score and performance using computer vision system (in Korean and English translation). J Korea Inst Electron Eng 22(5):429–435
35. Leplumey I, Camillerapp J, Lorette G (1993) A robust detector for music staves. In: Proceedings of the international conference on document analysis and recognition, Tsukuba Science city, Japan, pp 902–905
36. Luth N (2002) Automatic identification of music notations. In: Proceedings of the second international conference on WEB delivering of music (WEDELMUSIC), Darmstadt, Germany, pp 203–210
37. Macé S, Anquetil É, Coüasnon B (2005) A generic method to design pen-based systems for structured document composition: development of a musical score editor. In: Proceedings of the 1st workshop on improving and assessing pen-based input techniques, Edinburgh, pp 15–22
38. Mahoney J (1982) Automatic analysis of musical score images. B.sc. thesis, Massachussets Institute if technology, Dept of Engineering and Computer Science
39. Matsushima T, Ohteru S, Hashimoto S (1989) An integrated music information processing system: Psb-er. In: Proceedings of the international computer music conference, Columbus pp 191–198
40. Miyao H, Maruyama M (2007) An online handwritten music symbol recognition system. Int J Doc Anal Recognit 9(1):49–58
41. Miyao H, Nakano Y (1995) Head and stem extraction from printed music scores using a neural network approach. In: Proceedings of the 3rd international conference on document analysis and recognition, Montreal, Canada, pp 1074–1079
42. Modayur BR, Ramesh V, Haralick RM, Shapiro LG (1993) Muser: a prototype musical score recognition system using mathematical morphology. Mach Vis Appl 6(2–3):140–150
43. Ng K (2002) Music manuscript tracing. Blostein D, Kwon Y-B (eds) Graphics Recognition Algorithms and Applications, Volume 2390 of lecture notes in computer science, Springer, Berlin Heidelberg, pp 330–342.
44. Pinto J, Vieira P, Sosa J (2003) A new graph-like classification method applied to ancient handwritten musical symbols. Int J Doc Anal Recognit (IJDAR) 6(1):10–22
45. Prerau D (1970) Computer pattern recognition of standard engraved music notation. PhD thesis, Massachussets Institute if technology, Dept of Engineering and Computer Science
46. Pruslin D (1966) Automatic recognition of sheet music. PhD thesis, Massachussets Institute if technology
47. Pugin L (2006) Optical music recognition of early typographic prints using hidden markov models. In: International conference on music information retrieval, Victoria, Canada, pp 53–56
48. Pugin L, Burgoyne JA, Fujinaga I (2007) Goal-directed evaluation for the improvement of optical music recognition on early music prints. In: Rasmussen EM, Larson RR, Toms E, Sugimoto S (eds) Proceedings of the 7th ACM/IEEE-CS joint conference on digital libraries, Vancouver, Canada. ACM, pp 303–304
49. Pugin L, Hockman J, Burgoyne JA, Fujinaga I (2008) GAMERA versus ARUSPIX. Two optical music recognition approaches. In: Proceedings of the 9th international conference on music information retrieval, Philadelphia, USA, pp 419–424
50. Randriamahefa R, Cocquerez J, Fluhr C, Pépin F, Philipp S (1993) Printed music recognition. In: Proceedings of the international conference on document analysis and recognition, ICDAR, Tsukuba Science city, Japan, pp 898–901

51. Rebelo A, Capela G, Cardoso J (2010) Optical recognition of music symbols. Int J Doc Anal Recognit 13(1):19–31
52. Rebelo A, Fujinaga I, Paszkiewicz F, Marcal A, Guedes C, Cardoso J (2012) Optical music recognition: state-of-the-art and open issues. Int J Multimed Inf Retr 1(3):173–190
53. Roach J, Tatem J (1988) Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment. Pattern Recognit 21(1):33–44
54. Rossant F, Bloch I (2005) Optical music recognition based on a fuzzy modeling of symbol classes and music writing rules. In: IEEE international conference on image processing, Genova, Italy, vol 2, pp 538–541
55. Stückelberg MV, Doermann DS (1999) On musical score recognition using probabilistic reasoning. In: Proceedings of the international conference on document analysis and recognition, ICDAR, Bangalore, pp 115–118
56. Su B, Lu S, Pal U, Tan C (2012) An effective staff detection and removal technique for musical documents. In: IAPR international workshop on document analysis systems (DAS), Gold Coast, Queensland, Australia. IEEE Computer Society, pp 160–164
57. Toyama F, Shoji K, Miyamichi J (2006) Symbol recognition of printed piano scores with touching symbols. In: International conference on pattern recognition, Hong Kong, vol 2. IEEE Computer Society, pp 480–483

Further Reading

For further reading on Optical Music Recognition, the reader is referred to the critical survey by Blostein and Baird [5], as well as the book by Susan George [28]. Recently, Rebelo et al. have published [52] a survey which also includes a description about the available databases and software. Finally, and concerning the recognition of music symbols, Rebelo et al. have also compared the main techniques in [51].

In addition to the recognition of music scores, some researchers have been focused on writer identification. In this case, the goal is to identify the writer of a music score, which is especially useful for scholars when determining the authorship of anonymous historical documents. For further information, the reader is referred to [6, 22, 25, 29, 36].