TRENDS AND SURVEYS

Optical music recognition: state-of-the-art and open issues

**Ana Rebelo · Ichiro Fujinaga · Filipe Paszkiewicz ·
Andre R. S. Marcal · Carlos Guedes · Jaime S. Cardoso**

Abstract For centuries, music has been shared and remembered by two traditions: aural transmission and in the form of written documents normally called musical scores. Many of these scores exist in the form of unpublished manuscripts and hence they are in danger of being lost through the normal ravages of time. To preserve the music some form of typesetting or, ideally, a computer system that can automatically decode the symbolic images and create new scores is required. Programs analogous to optical character recognition systems called optical music recognition (OMR) systems have been under intensive development for many years. However, the results to date are far from ideal. Each of the proposed methods emphasizes different properties and therefore makes it difficult to effectively evaluate its competitive advantages. This article provides an overview of the literature concerning the automatic analysis of images of printed and handwritten musical scores. For self-containment and for the benefit of the reader, an introduction to OMR processing systems precedes the literature overview. The following study presents a reference scheme for any researcher wanting to compare new OMR algorithms against well-known ones.

A. Rebelo (✉) · F. Paszkiewicz · C. Guedes · J. S. Cardoso
FEUP, INESC Porto, Porto, Portugal
e-mail: arebelo@inescporto.pt

F. Paszkiewicz
e-mail: filipe.asp@gmail.com

C. Guedes
e-mail: carlosguedes@mac.com

J. S. Cardoso
e-mail: jaime.cardoso@inescporto.pt

I. Fujinaga
Schulich School of Music, McGill University, Montreal, Canada
e-mail: ich@music.mcgill.ca

A. R. S. Marcal
FCUP, CICGE, Porto, Portugal
e-mail: andre.marcal@fc.up.pt

1 Introduction

The musical score is the primary artifact for the transmission of musical expression for non-aural traditions. Over the centuries, musical scores have evolved dramatically in both symbolic content and quality of presentation. The appearance of musical typographical systems in the late nineteenth century and, more recently, the emergence of very sophisticated computer music manuscript editing and page-layout systems illustrate the continuous absorption of new technologies into systems for the creation of musical scores and parts. Until quite recently, most composers of all genres—film, theater, concert, sacred music—continued to use the traditional "pen and paper" finding manual input to be the most efficient. Early computer music typesetting software developed in the 1970s and 1980s produced excellent output but was awkward to use. Even the introduction of data entry from musical keyboard (MIDI piano for example) provided only a partial solution to the rather slow keyboard and mouse GUIs. There are many scores and parts still being "hand written". Thus, the demand for a robust and accurate optical music recognition (OMR) system remains.

Digitization has been commonly used as a possible tool for preservation, offering easy duplications, distribution, and digital processing. However, a machine-readable symbolic format from the music scores is needed to facilitate operations such as search, retrieval, and analysis. The manual transcription of music scores into an appropriate digital format

is very time consuming. The development of general image processing methods for object recognition has contributed to the development of several important algorithms for OMR. These algorithms have been central to the development of systems to recognize and encode music symbols for a direct transformation of sheet music into a machine-readable symbolic format.

The research field of OMR began with Pruslin [75] and Prerau [73] and, since then, has undergone much important advancements. Several surveys and summaries have been presented to the scientific community: Kassler [53] reviewed two of the first dissertations on OMR, Blostein and Baird [9] published an overview of OMR systems developed between 1966 and 1992, Bainbridge and Bell [3] published a generic framework for OMR (subsequently adopted by many researchers in this field), and both Homenda [47] and Rebelo et al. [83] presented pattern recognition studies applied to music notation. Jones et al. [51] presented a study in music imaging, which included digitalization, recognition, and restoration and also provided a well-detailed list of hardware and software in OMR together with an evaluation of three OMR systems.

Access to low-cost flat-bed digitizers during the late 1980s contributed to an expansion of OMR research activities. Several commercial OMR software have appeared, but none with a satisfactory performance in terms of precision and robustness, in particular for handwritten music scores [6]. Until now, even the most advanced recognition products including Notescan in Nightingale,[1] Midiscan in Finale,[2] Photoscore in Sibelius[3] and others such as Smartscore,[4] and Sharpeye[5] cannot identify all musical symbols. Furthermore, these products are focused primarily on recognition of typeset and printed music documents and while they can produce quite good results for these documents, they do not perform very well with hand-written music. The bi-dimensional structure of musical notation revealed by the presence of the staff lines alongside the existence of several combined symbols organized around the noteheads poses a high level of complexity in the OMR task.

In this paper, we survey the relevant methods and models in the literature for the optical recognition of musical scores. We address only offline methods (page-based imaging approaches), although the current proliferation of small electronic devices with increasing computation power, such as tablets, smartphones, may increase the interest in online methods, these are out of the scope of this paper. In Sect. 1.1 of this introductory section a description of a

typical architecture of an OMR system is given. Section 1.2, which addresses the principal properties of the music symbols, completes this introduction. The image preprocessing stage is addressed in Sect. 2. Several procedures are usually applied to the input image to increase the performance of the subsequent steps. In Sects. 3 and 4, a study of the state of the art for the music symbol detection and recognition is presented. Algorithms for detection and removal of staff lines are also presented. An overview of the works done in the fields of musical notation construction and final representation of the music document is made in Sect. 5. Existing standard datasets and performance evaluation protocols are presented in Sect. 6. Section 7 states the open issues in handwritten music scores and the future trends in the OMR using this type of scores. Section 8 concludes this paper.

1.1 OMR architecture

Breaking down the problem of transforming a music score into a graphical music-publishing file in simpler operations is a common but complex task. This is consensual among most authors that work in the field. In this paper we use the framework outlined in [83].

The main objectives of an OMR system are the recognition, the representation and the storage of musical scores in a machine-readable format. An OMR program should thus be able to recognize the musical content and make the semantic analysis of each musical symbol of a music work. In the end, all the musical information should be saved in an output format that is easily readable by a computer.

A typical framework for the automatic recognition of a set of music sheets encompasses four main stages (see Fig. 1):

1. image preprocessing;
2. recognition of musical symbols;
3. reconstruction of the musical information in order to build a logical description of musical notation; and
4. construction of a musical notation model to be represented as a symbolic description of the musical sheet.

For each of the stages described above, different methods exist to perform the respective task.

In the image preprocessing stage, several techniques—e.g., enhancement, binarization, noise removal, blurring, deskewing—can be applied to the music score to make the recognition process more robust and efficient. The reference lengths staff line thickness (staffline_height) and vertical line distance within the same staff (staffspace_height) are often computed, providing the basic scale for relative size comparisons (Fig. 5).

The output of the image preprocessing stage constitutes the input for the next stage, the recognition of musical sym-

---

[1] http://www.ngale.com/.

[2] http://www.finalemusic.com/.

[3] http://www.neuratron.com/photoscore.htm.

[4] http://www.musitek.com/.

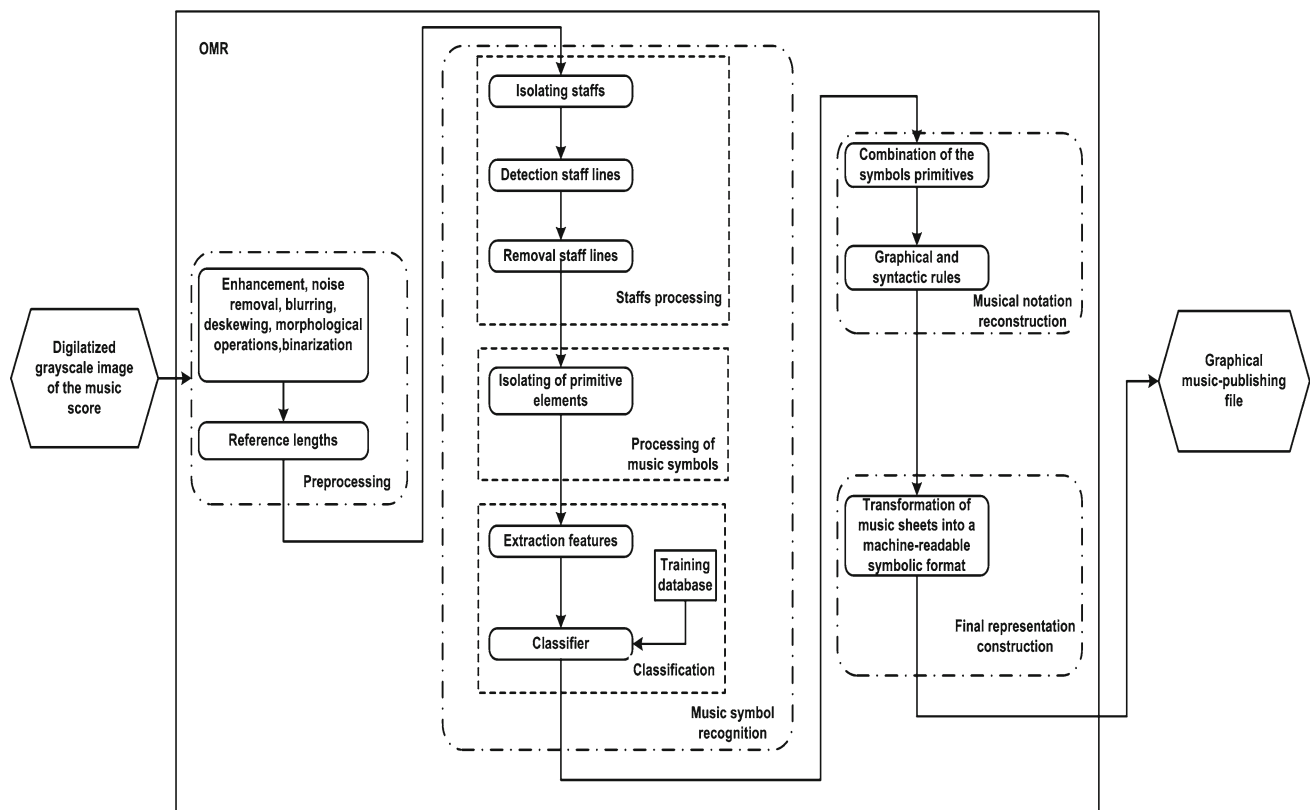[5] http://www.music-scanning.com/.

Fig. 1 Typical architecture of an OMR processing system

bols. This is typically further subdivided into three parts: (1) staff line detection and removal, to obtain an image containing only the musical symbols; (2) symbol primitive segmentation; and (3) symbol recognition. In this last stage the classifiers usually receive raw pixels as input features. However, some works also consider higher-level features, such as information about the connected components or the orientation of the symbol.

Classifiers are built by taking a set of labeled examples of music symbols and randomly split them into training and test sets. The best parameterization for each model is normally found based on a cross validation scheme conducted on the training set.

The third and fourth stages (musical notation reconstruction and final representation construction) can be intrinsically intertwined. In the stage of musical notation reconstruction, the symbol primitives are merged to form musical symbols. In this step, graphical and syntactic rules are used to introduce context information to validate and solve ambiguities from the previous module (music symbol recognition). Detected symbols are interpreted and assigned a musical meaning. In the fourth and final stages (final representation construction), a format of musical description is created with the previously produced information. The system
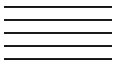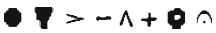
output is a graphical music-publishing file, like MIDI or MusicXML.

Some authors use several algorithms to perform different tasks in each stage, such as using an algorithm for detecting noteheads and a different one for detecting the stems. For example, Byrd and Schindele [13] and Knopke and Byrd [55] use a voting system with a comparison algorithm to merge the best features of several OMR algorithms to produce better results.

1.2 Properties of the musical symbols

Music notation emerged from the combined and prolonged efforts of many musicians. They all hoped to express the essence of their musical ideas by written symbols [80]. Music notation is a kind of alphabet, shaped by a general consensus of opinion, used to express ways of interpreting a musical passage. It is the visual manifestation of interrelated properties of musical sound such as pitch, dynamics, time, and timbre. Symbols indicating the choice of tones, their duration, and the way they are performed are important because they form this written language that we call music notation [81]. In Table 1, we present some common Western music notation symbols.

Table 1 Music notation

Symbols	Description
	Staff: An arrangement of parallel lines, together with the spaces between them
	Treble, Alto, and Bass clef: The first symbols that appear at the beginning of every music staff and tell us which note is found on each line or space
	Sharp, Flat and Natural: The signs that are placed before the note to designate changes in sounding pitch
	Beams: Used to connect notes in note-groups; they demonstrate the metrical and the rhythmic divisions
	Staccato, Staccatissimo, Dynamic, Tenuto, Marcato, Stopped note, Harmonic and Fermata: Symbols for special or exaggerated stress upon any beat, or portion of a beat
	Quarter, Half, Eighth, Sixteenth, Thirty-second and Sixty-fourth notes: The Quarter note (closed notehead) and Half note (open notehead) symbols indicate a pitch and the relative time duration of the musical sound. Flags (e.g. Eighth note) are employed to indicate the relative time values of the notes with closed noteheads
	Quarter, Eighth, Sixteenth, Thirty-second and Sixty-fourth rests: These indicate the exact duration of silence in the music; each note value has its corresponding rest sign; the written position of a rest between two barlines is determined by its location in the meter
	Ties and Slurs: Ties are a notational device used to prolong the time value of a written note into the following beat. The tie appears to be identical to slur, however, while tie almost touches the notehead center, the slur is set somewhat above or below the notehead. Ties are normally employed to join the time value of two notes of identical pitch; Slurs affect note-groups as entities indicating that the two notes are to be played in one physical stroke, without a break between them
	Mordent and Turn: Ornaments symbols that modify the pitch pattern of individual notes

Improvements and variations in existing symbols, or the creation on new ones, came about as it was found necessary to introduce a new instrumental technique, expression or articulation. New musical symbols are still being introduced in modern music scores, to specify a certain technique or gesture. Other symbols, especially those that emerged from extended techniques, are already accepted and known by many musicians (e.g. microtonal notation) but are still not available in common music notation (CMN) software. Musical notation is thus very extensive if we consider all the existing possibilities and their variations.

Moreover, the wider variability of the objects (in size and shape), found on handwritten music scores, makes the operation of music symbol extraction one of the most complex and difficult in an OMR system. Publishing variability in handwritten scores is illustrated in Fig. 2. In this example, we can see that for the same clef symbol and beam symbol we may have different thicknesses and shapes.

Fig. 2 Variability in handwritten music scores

2 Image preprocessing

The music scores processed by the state-of-art algorithms, described in the following sections, are mostly written in

Fig. 3 Some examples of music scores used in the state-of-art algorithms. **a** From Rebelo [81, Fig. 4.4a]

(a) Modern music notation.

(b) Early music notation.
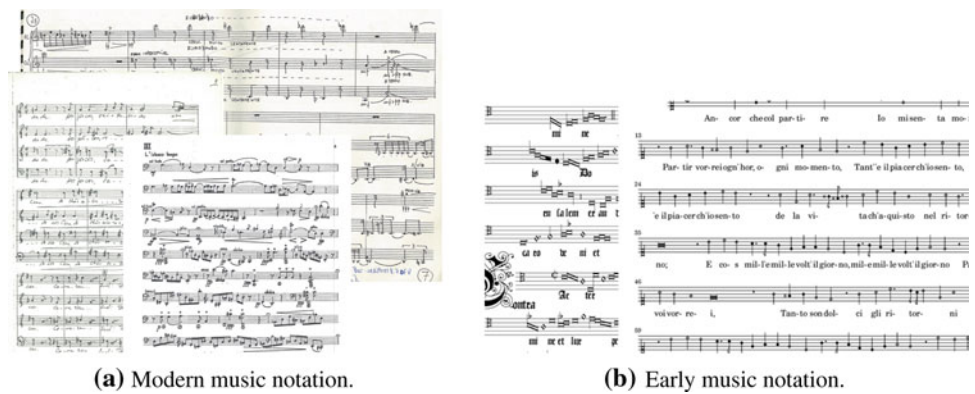
a standard modern notation (from the twentieth century). However, there are also some methods proposed for sixteenth and seventeenth century printed music. Figure 3 shows typical music scores used for the development and testing of algorithms in the scientific literature. In most of the proposed works, the music sheets were scanned at a resolution of 300 dpi [16,26,35,37,45,55,64,83,89]. Other resolutions were also considered: 600 dpi [56,87] or 400 dpi [76,96]. No studies have been carried out to evaluate the dependency of the proposed methods on other resolution values, thus restricting the quality of the objects presented in the music scores, and consequently the performance of all OMR algorithms.

In digital image processing, as in all signal processing systems, different techniques can be applied to the input, making it ready for the detection steps. The motivation is to obtain a more robust and efficient recognition process. Enhancement [45], binarization (e.g. [16,35,41,43,45,64,98]), noise removal (e.g. [41,45,96,98]), blurring [45], de-skewing (e.g. [35,41,45,64,98]), and morphological operations [45] are the most common techniques for preprocessing music scores.

2.1 Binarization

Almost all OMR systems start with a binarization process. This means that the digitalized image must be analyzed to determine what is useful (the objects, being the music symbols and staves) and what is not (the background, noise). To make binarization an automatic process, many algorithms have been proposed in the past, with different success rates, depending on the problem at hand. Binarization has the big virtue in OMR of facilitating the following tasks by reducing the amount of information they need to process. In turn, this results in higher computational efficiency (more important in the past than nowadays) and eases the design of models to tackle the OMR task. It has been easier to propose algorithm for line detection, symbol segmentation,

and recognition in binary images than in grayscale or color images. This approach is also supported by the typical binary nature of music scores. Usually, the author does not aim to portray information in the color; it is more a consequence of the writing or of the digitalization process. However, since binarization often introduces artifacts, it is not clear the advantages of binarization in the complete OMR process.

Burgoyne et al. [12] and Pugin et al. [77] presented a comparative evaluation of image binarization algorithms applied to sixteenth-century music scores. Both works used Aruspix, a software application for OMR which provides symbol-level recall and precision rate to measure the performance of different binarization procedures. In [12] they worked with a set of 8,000 images. The best result was obtained with the Brink and Pendock [10]'s method. The adaptive algorithm with the highest ranking was Gatos et al. [42]. Nonetheless, the binarization of the music score still needs attention with researchers invariably using standard binarization procedures, such as the Otsu's method (e.g. [16,45,76,83]). The development of binarization methods specific to music scores potentially shows performances that are better than the generic counterparts', and leverages the performance of subsequent operations [72].

The fine-grained categorization of existing techniques presented in Fig. 4 follows the survey in [92], where the classes were chosen according to the information extracted from the image pixels. Despite this labeling the categories are essentially organized into two main topics: global and adaptive thresholds.

Global thresholding methods apply one threshold to the entire image. Ng and Boyle [66] and Ng et al. [68] have adopted the technique developed by Ridler and Calvard [86]. This iterative method achieves the final threshold through an average of two sample means ($T = (\mu_b + \mu_f)/2$). Initially, a global threshold value is selected for the entire image and then a mean is computed for the background pixels (μ_b) and for the foreground pixels (μ_f). The process is repeated based on the new threshold computed from μ_b and μ_f, until the
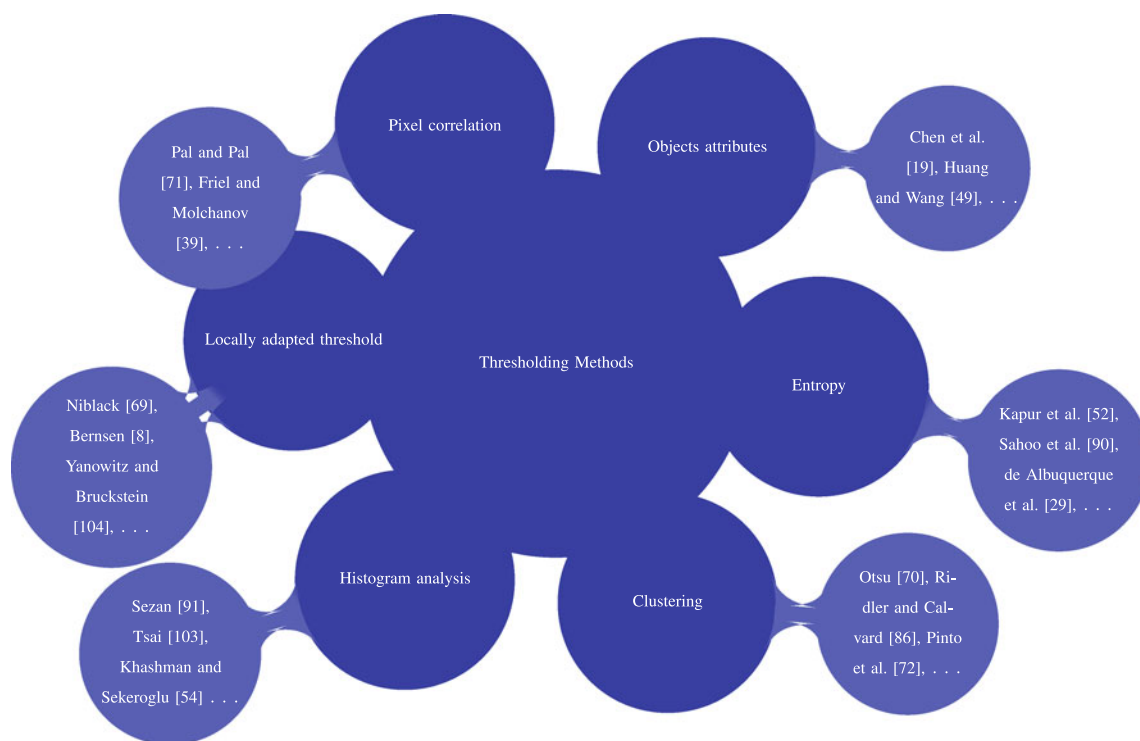
Fig. 4 Landscape of automated thresholding methods. From Pinto et al. [72, Fig. 1]

threshold value does not change any more. According to [101, 102], Otsu's procedure is ranked as the best and the fastest of these methods [70]. In the OMR field, several research works have used this technique [16,45,76,83,98].

In adaptive binarization methods, a threshold is assigned to each pixel using local information from the image. Consequently, the global thresholding techniques can extract objects from uniform backgrounds at a high speed, whereas the local thresholding methods can eliminate dynamic backgrounds although with a longer processing time. One of the most used methods is Niblack [69]'s method which uses the mean and the standard deviation of the pixel's vicinity as local information for the threshold decision. The research work carried out by [36,37,96] applied this technique to their OMR procedures.

Only recently the domain knowledge has been used at the binarization stage in the OMR area. The work presented in [72] proposes a new binarization method which not only uses the raw pixel information, but also considers the image content. The process extracts content-related information from the grayscale image, the staff line thickness (staff-line_height), and the vertical line distance within the same staff (staffspace_height), to guide the binarization procedure. The binarization algorithm was designed to maximize the number of pairs of consecutive runs summing staffline_height + staffspace_height. The authors suggest that this maximization increases the quality of the binarized lines

Fig. 5 The characteristic page dimensions of staffline_height and staffspace_height. From Cardoso and Rebelo [17]

and consequently the subsequent operations in the OMR system.

Until now Pinto et al. [72] seems to be the only threshold method that uses content of gray-level images of music scores deliberately to perform the binarization.

2.2 Reference lengths

In the presence of a binary image most OMR algorithms rely on an estimation of the staff line thickness and the distance that separates two consecutive staff lines—see Fig. 5.

Further processing can be performed based on these values and be independent of some predetermined magic numbers. The use of fixed threshold numbers, as found in other areas, causes systems to become inflexible, making it more difficult for them to adapt to new and unexpected situations.

The well-known run-length encoding (RLE), which is a very simple form of data compression in which runs of data

Fig. 6 Example of an image where the estimation of staffline_height and staffspace_height by vertical runs fails. From Cardoso and Rebelo [17, Fig. 2]
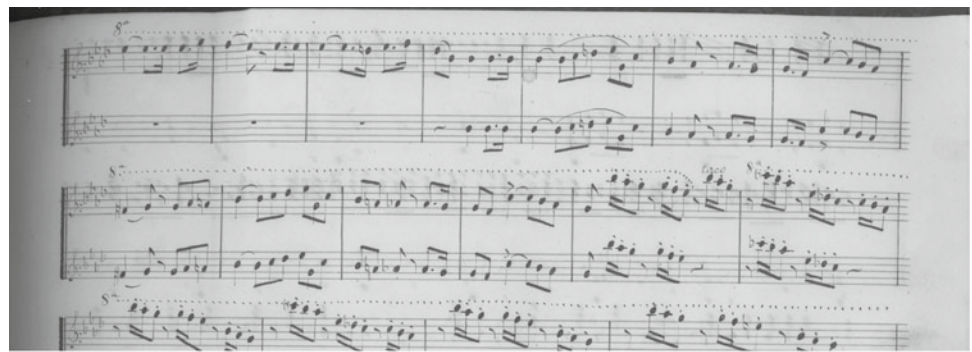
(a) Original music score #17.

(b) Score binarized with Otsu's method.

are represented as a single data value and count, is often used to determine these reference values (e.g. [16,26,32,41, 89]) —the other technique can be found in [98]. In a binary image, used here as input for the recognition process, there are only two values: one and zero. In such a case, the RLC is even more compact, because only the lengths of the runs are needed. For example, the sequence {1 1 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1} can be coded as 2, 1, 3, 2, 4, 2, 4, 1, 5, 2, 6, assuming that 1 starts a sequence (if a sequence starts with a 0, the length of zero would be used). By encoding each column of a digitized score using RLE, the most common black-run represents the staffline_height and the most common white-run represents the staffspace_height.

Nonetheless, there are music scores with high levels of noise, not only because of the low quality of the original paper in which it is written, but also because of the artifacts introduced during digitalization and binarization. These aspects make the results unsatisfactory, impairing the quality of subsequent operations. Figure 6 illustrates this problem. For this music score, we have pale staff lines that broke up during binarization providing the conventional estimation staffline_height = 1 and staffspace_height = 1 (the true values are staffline_height = 5 and staffspace_height = 19).

The work suggested by Cardoso and Rebelo [17], which encouraged the work proposed in [72], presents a more robust estimation of the sum of staffline_height and staffspace_ height by finding the most common sum of two consecutive
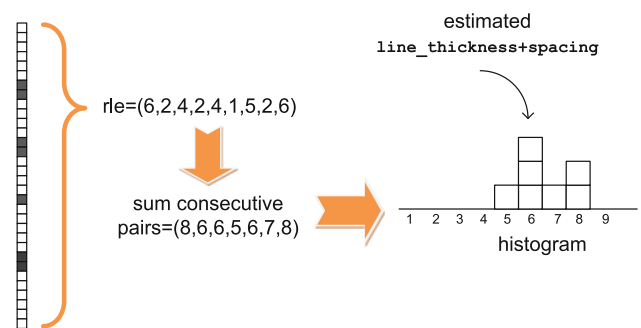
Fig. 7 Illustration of the estimation of the reference value staffline_height and staffspace_height using a single column. From Pinto et al. [72, Fig. 2]

vertical runs (either black run followed by white run or the reverse). The process is illustrated in Fig. 7.

In this manner, to reliably estimate staffline_height and staffspace_height values, the algorithm starts by computing the 2D histogram of the pairs of consecutive vertical runs and afterwards it selects the most common pair for which the sum of the runs equals staffline_height + staffspace_height.

3 Staff line detection and removal

Staff line detection and removal are fundamental stages in many OMR systems. The reason to detect and remove the staff lines lies on the need to isolate the musical symbols for a

more efficient and correct detection of each symbol present in the score. Notwithstanding, there are authors who suggested algorithms without the need to remove the staff lines [5,7,45, 58,68,76,93]. In here, the decision is between simplification to facilitate the following tasks with the risk of introducing noise. For instance, symbols are often broken in this process, or bits of lines that are not removed are interpreted as part of symbols or new symbols. The issue will always be related to the preservation of as much information as possible for the next task, with the risk of increasing computational demand and the difficulty of modeling the data.

Staff detection is complicated due to a variety of reasons. Although the task of detecting and removing staff lines is completed fairly accurately in some OMR systems, it still represents a challenge. The distorted staff lines are a common problem in both printed and handwritten scores. The staff lines are often not straight or horizontal (due to wrinkles or poor digitization) and in some cases hardly parallel to each other. Moreover, most of these works are old, which means that the quality of the paper and ink has decreased severely. Another interesting setting is the common modern case where music notation is handwritten on paper with preprinted staff lines.

The simplest approach consists of finding local maxima on the horizontal projection of the black pixels of the image [41,79]. Assuming straight and horizontal lines, these local maxima represent line positions. Several horizontal projections can be made with different image rotation angles, keeping the image where the local maximum is higher. This eliminates the assumption that the lines are always horizontal. Miyao and Nakano [62] use Hough Transform to detect staff lines. An alternative strategy for identifying staff lines is to use vertical scan lines [18]. This process is based on a line adjacency graph (LAG). LAG searches for potential sections of lines: sections that satisfy criteria related to aspect ratio, connectedness, and curvature. More recent works present a sophisticated use of projection techniques combined to improve the basic approach [2,5,7,89].

Fujinaga [41] incorporates a set of image processing techniques in the algorithm, including run-length coding (RLC), connected-component analysis, and projections. After applying the RLC to find the thickness of staff lines and the space between the staff lines, any vertical black run that is more than twice the staff line height is removed from the original. Then, the connected components are scanned to eliminate any component whose width is less than the staff space height. After a global de-skewing, taller components, such as slurs and dynamic wedges are removed.

Other techniques for finding staff lines include the grouping of vertical columns based on their spacing, thickness, and vertical position on the image [85], rule-based classification of thin horizontal line segments [60], and line tracing [73,88,98]. The methods proposed in [63,95] operate on a

set of *staff segments*, with methods for linking two segments horizontally and vertically and merging two overlapped segments. Dutta et al. [32] proposed a similar but simpler procedure than previous ones. The authors considered a staff line segment as an horizontal connection of vertical black runs with uniform height and validating it using neighboring properties. The work by Dalitz et al. [26] is an improvement on the methods of [63,95].

In spite of the variety of methods available for staff lines detection, they all have some limitations. In particular, lines with some curvature or discontinuities are inadequately resolved. The dash detector [57] is one of a few works that try to handle discontinuities. The dash detector is an algorithm that searches the image, pixel by pixel, finding black pixel regions that it classifies as stains or dashes. Then, it tries to unite the dashes to create lines.

A common problem to all the aforementioned techniques is that they try to build staff lines from local information, without properly incorporating global information in the detection process. None of the methods tries to define a reasonable process from the intrinsic properties of staff lines, namely the fact that they are the only extensive black objects on the music score. Usually, the most interesting techniques arise when one defines the detection process as the result of optimizing some global function. In [16], the authors proposed a graph-theoretic framework where the staff line is the result of a global optimization problem. The new staff line detection algorithm suggests using 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. As staff lines are almost the only extensive black objects on the music score, the path to look for is the shortest path between the two margins if paths (almost) entirely through black pixels are favored. The performance was experimentally supported on two test sets adopted for the qualitative evaluation of the proposed method: the test set of 32 synthetic scores from [26], where several known deformations were applied, and a set of 40 real handwritten scores, with ground truth obtained manually.

4 Symbol segmentation and recognition

The extraction of music symbols is the operation following the staff line detection and removal. The segmentation process consists of locating and isolating the musical objects to identify them. In this stage, the major problems in obtaining individual meaningful objects are caused by printing and digitalization, as well as paper degradation over time. The complexity of this operation concerns not only the distortions inherent to staff lines, but also broken and

overlapping symbols, differences in sizes, and shapes and zones of high density of symbols. The segmentation and classification process has been the object of study in the research community (e.g. [5,20,89,100]).

The most usual approach for symbol segmentation is a hierarchical decomposition of the music image. A music sheet is first analyzed and split by staffs and then the elementary graphic symbols are extracted: noteheads, rests, dots, stems, flags, etc. (e.g. [20,31,45,62,66,83,85,98]). Although in some approaches [83] noteheads are joined with stems and also with flags for the classification phase, in the segmentation step these symbols are considered to be separate objects. In this manner, different methods use equivalent concepts for primitive symbols.

Usually, the primitive segmentation step is made along with the classification task [89,100]; however, there are exceptions [5,7,41]. Mahoney [60] builds a set of candidates to one or more symbol types and then uses descriptors to select the matching candidates. Carter [18] and Dan [28] use a LAG to extract symbols. The objects resulting from this operation are classified according to the bounding box size, the number, and organization of their constituent sections. Reed and Parker [85] also uses LAGs to detect lines and curves. However, accidentals, rests and clefs are detected by a character profile method, which is a function that measures the perpendicular distance of the object's contour to reference axis, and noteheads are recognized by template matching. Other authors have chosen to apply projections to detect primitive symbols [5,7,41,74]. The recognition is done using features extracted from the projection profiles. In [41], the k-nearest neighbor rule is used in the classification phase, while neural networks is the classifier selected in [5,7,62,66]. Choudhury et al. [20] proposed the extraction of symbol features, such as width, height, area, number of holes, and low-order central moments, whereas Taubman [99] preferred to extract standard moments, centralized moments, normalized moments, and Hu moments. Both systems classify the music primitives using the k-nearest neighbor method.

Randriamahefa et al. [79] proposed a structural method based on the construction of graphs for each symbol. These are isolated using a region-growing method and thinning. In [89] a fuzzy model supported on a robust symbol detection and template matching was developed. This method is set to deal with uncertainty, flexibility, and fuzziness at the level of the symbol. The segmentation process is addressed in two steps: individual analysis of musical symbols and fuzzy model. In the first step, the vertical segments are detected by a region-growing method and template matching. The beams are then detected by a region-growing algorithm and a modified Hough Transform. The remaining symbols are extracted again by template matching. As a result of this first step, three recognition hypotheses occur, and the fuzzy model is then used to make a consistent decision.

Other techniques for extracting and classifying musical symbols include rule-based systems to represent the musical information, a collection of processing modules that communicate by a common working memory [88] and pixel tracking with template matching [100]. Toyama et al. [100] check for coherence in the primitive symbols detected by estimating overlapping positions. This evaluation is carried out using music writing rules. Coüasnon [21,23] proposed a recognition process entirely controlled by grammar which formalizes the musical knowledge. Bainbridge [2] uses PRIMitive Expression LAnguage (PRI-MELA) language, which was created for the CANTerbury OMR (CANTOR) system, to recognize primitive objects. In [85] the segmentation process involves three stages: line and curves detection by LAGs, accidentals, rests, and clefs detection by a character profile method and noteheads recognition by template matching. Fornés et al. [34] proposed a classifier procedure for handwritten symbols using the Adaboost method with a blurred shape model descriptor.

It is worth mentioning that in some works, we assist to a new line of approaches that avoid the prior segmentation phase in favor of methods that simultaneously segment and recognize. In [76,78] the segmentation task is based on Hidden Markov models (HMMs). This process performs segmentation and classification simultaneously. The extraction of features directly from the image frames has advantages. Particularly, it avoids the need to segment and track the objects of interest, a process with a high degree of difficulty and prone to errors. However, this work applied this technique only in very simple scores, that is, scores without slurs or more than one symbol in the same column and staff.

In [68] a framework based on a mathematical morphological approach commonly used in document imaging is proposed. The authors applied a skeletonization technique with an edge detection algorithm and a stroke direction operation to segment the music score. Goecke [45] applies template matching to extract musical symbols. In [99] the symbols are recognized using statistical moments. This way, the proposed OMR system is trained with strokes of musical symbols and a statistical moment is calculated for each one of them; the class for an unknown symbol is assigned based on the closest match. In [35] the authors start by using median filters with a vertical structuring element to detect vertical lines. Then they apply a morphological opening using an elliptical structuring element to detect noteheads. The bar lines are detected considering its height and the absence of noteheads in its extremities. Clef symbols are extracted using Zernike moments and Zoning, which code shapes based on the statistical distribution of points. Although a good performance was verified in the detection of these specific symbols, the authors did not extract the other symbols that were also present on a music score and are indispensable for a complete optical music recognition. In [83] the segmentation of the objects is

based on an hierarchical decomposition of a music image. A music sheet is first analyzed and split by staffs. Subsequently, the connected components are identified. To extract only the symbols with appropriate size, the connected components detected in the previous step are selected. Since a bounding box of a connected component can contain multiple connected components, care is taken to avoid duplicate detections or failure to detect any connected component. In the end, all music symbols are extracted based on their shape. In [98] the symbols are extracted using a connected component process and small elements are removed based on their size and position on the score. The classifiers adopted were the kNN, the Mahalanobis distance, and the Fisher discriminant.

Some studies were conducted in the music symbols classification phase, more precisely the comparison of results between different recognition algorithms. Homenda and Luckner [48] studied decision trees and clustering methods. The symbols were distorted by noise, printing defects, different fonts, skew and curvature of scanning. The study starts with the extraction of some symbols features. Five classes of music symbols were considered. Each class had 300 symbols extracted from 90 scores. This investigation encompassed two different classification approaches: classification with and without rejection. In the later case, every symbol belongs to one of the given classes, while in the classification with rejection, not every symbol belongs to a class. Thus, the classifier should decide if the symbol belongs to a given class or if it is an extraneous symbol and should not be classified. Rebelo et al. [83] carried out an investigation on four classification methods, namely support vector machines (SVMs), neural networks (NNs), nearest neighbor (kNN) and Hidden Markov Models. The performances of these methods were compared using both real and synthetic scores. The real scores consisted of a set of 50 handwritten scores from 5 different musicians, previously binarized. The synthetic data set included 18 scores (considered to be ideal) from different publishers to which known deformations have been applied: rotation and curvature. In total, 288 images were generated from the 18 original scores. The full set of training patterns extracted from the database of scores was augmented with replicas of the existing patterns, transformed according to the elastic deformation technique [50]. Such transformations tried to introduce robustness in the prediction regarding the known variability of symbols. Fourteen classes were considered with a total of 3,222 handwritten music symbols and 2,521 printed music symbols. In the classification, the SVMs, NNs, and kNN received raw pixels as input features (a 400 feature vector, resulting from a 20×20 pixel image); the HMM received higher-level features, such as information about the connected components in a 30×150 pixel window. The SVMs attained the best performance while the HMMs had the worse result. The use of elastic deformations did not improve the performance of the classifiers. Three explanations for this outcome were suggested: the distortions created were not the most appropriate, the data set of symbols was already diverse, or the adopted features were not proper for this kind of variation.

A more recent procedure for pattern recognition is the use of classifiers with a reject option [25,46,94]. The method integrates a confidence measure in the classification model to reject uncertain patterns, namely broken and touching symbols. The advantage of this approach is the minimization of misclassification errors in the sense that it chooses not to classify certain symbols (which are then manually processed).

Lyrics recognition is also an important issue in the OMR field, since lyrics make the music document even more complex. In [11] techniques for lyric editor and lyric lines extraction were developed. After staff lines removal, the authors computed baselines for both lyrics and notes, stressing that baselines for lyrics would be highly curved and undulating. The baselines are extracted based on local minima of the connected components of the foreground pixels. This technique was tested on a set of 40 images from the Digital Image Archive of Medieval Music. In [44] an overview of existing solutions to recognize the lyrics in Christian music sheets is described. The authors stress the importance of associating the lyrics with notes and melodic parts to provide more information to the recognition process. Resolutions for page segmentation, character recognition, and final representation of symbols are presented.

Despite the number of techniques already available in the literature, research on improving symbol segmentation and recognition is still important and necessary. All OMR systems depend on this step.

5 Musical notation construction and final representation

The final stage in a music notation construction engine is to extract the musical semantics from the graphically recognized shapes and store them in a musical data structure. Essentially, this involves combining the graphically recognized musical features with the staff systems to produce a musical data structure representing the meaning of the scanned image. This is accomplished by interpreting the spatial relationships between the detected primitives found in the score. If we are dealing with optical character recognition (OCR) this is a simple task, because the layout is predominantly one-dimensional. However, in music recognition, the layout is much more complex. The music is essentially two dimensional, with pitch represented vertically and time horizontally. Consequently, positional information is extremely important. The same graphical shape can mean different things in different situations. For instance, to determine if a curved line between two notes is a slur or a tie, it is

necessary to consider the pitch of the two notes. Moreover, musical rules involve a large number of symbols that can be spatially far from each other in the score.

Several research works have suggested the introduction of the musical context in the OMR process by a formalization of musical knowledge using a grammar (e.g. [4,7,22,73,74, 85]). The grammar rules can play an important role in music creation. They specify how the primitives are processed, how a valid musical event should be made, and even how graphical shapes should be segmented. Andronico and Ciampa [1] and Prerau [74] were pioneers in this area. One of Fujinaga's first works focused on the characterization of music notation by means of a *context-free* and $LL(k)$ grammar. Coüasnon [22,24] also based their works on a grammar, which is essentially a description of the relations between the graphical objects and a parser, which is the introduction of musical context with syntactic or semantic information. The author claims that this approach will reduce the risk of generating errors imposed during the symbols extraction, using only very local information. The proposed grammar is implemented in λProlog, a higher dialect of Prolog with more expressive power, with semantic attributes connected to *C* libraries for pattern recognition and decomposition. The grammar is directly implemented in λProlog using definite clause grammars (DCG's) techniques. It has two levels of parsing: a graphical one corresponding to the physical level and a syntactic one corresponding to the logical level. The parser structure is a list composed of segments (non-labeled) and connected components, which do not necessarily represent a symbol. The first step of the parser is the labeling process and the second is the error detection. Both operations are supported by the context introduced in the grammar. However, no statistical results are available for this system.

Bainbridge [2] also implemented a grammar-based approach using DCG's to specify the relationships between the recognized musical shapes. This work describes the CANTOR system, which has been designed to be as general as possible by allowing the user to define the rules that describe the music notation. Consequently, the system is readily adaptable to different publishing styles in CMN. The authors argue that their method overcame the complexity imposed in the parser development operation proposed in [22,24]. CANTOR avoids such drawbacks by using a *bag*[6] of tokens instead of using a *list* of tokens. For instance, instead of getting a unique next symbol, the grammar can "request" a token, e.g. a notehead, from the bag, and if its position does not fit in with the current musical feature that is being parsed, then the grammar can backtrack and request the "next" notehead from the bag. To deal with complexity time,

the process uses derivation trees of the assembled musical features during the parse execution. In a more recent work Bainbridge and Bell [4] incorporated a basic graph in CANTOR system according to each musical feature's position (x, y). The result is a lattice-like structure of musical feature nodes that are linked horizontally and vertically. This final structure is the musical interpretation of the scanned image. Consequently, additional routines can be incorporated in the system to convert this graph into audio application files (such as MIDI and CSound) or music editor application files (such as Tilia or NIFF).

Prerau [73] makes a distinction between notational grammars and higher-level grammars for music. While notation grammars allow the computer to recognize important music relationships between the symbols, the higher-level grammars deal with phrases and larger units of music.

Other techniques to construct the musical notation are based on fusion of musical rules and heuristics (e.g. [28,31, 68,89]) and common parts on the row and column histograms for each pair of symbols [98]. Rossant and Bloch [89] proposed an OMR system with two stages: detection of the isolated objects and computation of hypotheses, both using low-level preprocessing, and final correct decision based on high-level processing which includes contextual information and music writing rules. In the graphical consistency (low-level processing) stage, the purpose is to compute the compatibility degree between each object and all the surrounding objects, according to their classes. The graphical rules used by the authors were

– Accidentals and notehead: an accidental is placed before a notehead and at same height.
– Noteheads and dots: the dot is placed after or above a notehead in a variable distance.
– Between any other pair of symbols: they cannot overlap.

In the syntactic consistency (high-level processing) stage, the aim is to introduce rules related to tonality, accidentals, and meter. Here, the key signature is a relevant parameter. This group of symbols is placed in the score as an ordered sequence of accidentals placed just after the clef. In the end, the score meter (number of beats per bar) is checked. In [65,67,68] the process is also based on a low- and high-level approaches to recognize music scores. Once again, the reconstruction of primitives is done using basic musical syntax. Therefore, extensive heuristics and musical rules are applied to reconfirm the recognition. After this operation, the correct detection of key and time signature becomes crucial. They provide a global information about the music score that can be used to detect and correct possible recognition errors. The developed system also incorporates a module to output the result into a expMIDI (expressive MIDI) format. This was an attempt to surmount the limitations of MIDI for

---

[6] A bag is a one-dimensional data structure which is a cross between a list and a set; it is implemented in Prolog as a predicate that extracts elements from a list, with unrestricted backtracking.

expressive symbols and other notations details, such as slurs and beaming information.

More research works produced in the past use abductive constraint logic programming (ACLP) [33] and sorted lists that connect all inter-related symbols [20]. In [33] an ACLP system, which integrates into a single framework abductive logic programming (ALP) and constraint logic programming (CLP), is proposed. This system allows feedback between the high-level phase (musical symbols interpretation) and the low-level phase (musical symbols recognition). The recognition module is carried out through object feature analysis and graphical primitive analysis, while the interpretation module is composed of music notation rules to reconstruct the music semantics. The system output is a graphical music-publishing file, like MIDI. No practical results are known for this architecture's implementation.

Other procedures try to automatically synchronize sheet music scanned with a corresponding CD audio recording [27,38,56] using a matching between OMR algorithms and digital signal processing. Based on an automated mapping procedure, the authors identify scanned pages of music score by means of a given audio collection. Both scanned score and audio recording are turned into a common mid-level representation—chroma-based features, where the *chroma* corresponds to the 12 traditional pitch classes of the equal-tempered scale—whose sequences are time-aligned using algorithms based on dynamic time warping (DTW). In the end, a combination of this alignment with OMR results is performed to connect spatial positions within audio recording to regions within scanned images.

5.1 Summary

Most notation systems make it possible to import and export the final representation of a musical score for MIDI. However, several other music encoding formats for music have been developed over the years—see Table 2. The used OMR systems are non-adaptive and consequently they do not improve their performance through usage. Studies have been carried out to overcome this limitation by merging multiple OMR systems [13,55]. Nonetheless, this remains a challenge. Furthermore, the results of the most OMR systems are only for the recognition of printed music scores. This is the major gap in state-of-the-art frameworks. With the exception for PhotoScore, which works with handwritten scores, most of them fail when the input image is highly degraded such as photocopies or documents with low-quality paper. The work developed in [14] is the beginning of a web-based system that will provide broad access to a wide *corpus* of handwritten unpublished music encoded in digital format. The system includes an OMR engine integrated with an archiving system and a user-friendly interface for searching, browsing, and editing. The output of digitized scores is stored in MusicXML

Table 2 The most relevant OMR software and programs

Software and program	Output file
SmartScore[a]	Finale, MIDI, NIFF, PDF
SharpEye[b]	MIDI, MusicXML, NIFF
PhotoScore[c]	MIDI, MusicXML, NIFF, PhotoScore, WAVE
Capella-Scan[d]	Capella, MIDI, MusicXML
ScoreMaker[e]	MusicXML
Vivaldi Scan[f]	Vivaldi, XML, MIDI
Audiveris[g]	MusicXML
Gamera[h]	XML files

[a] http://www.musitek.com/
[b] http://www.music-scanning.com/
[c] http://www.neuratron.com/photoscore.htm
[d] http://www.capella-software.com/capella-scan.cfm
[e] http://www.music-notation.info/en/software/SCOREMAKER.html
[f] http://www.vivaldistudio.com/Eng/VivaldiScan.asp
[g] http://audiveris.kenai.com/
[h] http://gamera.informatik.hsnr.de/

which is a recent and expanding music interchange format designed for notation, analysis, retrieval, and performance applications.

6 Available datasets and performance evaluation

There are some available datasets that can be used by OMR researchers to test the different steps of an OMR processing system. Pinto et al. [72] made available the code and the database[7] they created to estimate the results of binarization procedures in the preprocessing stage. This database is composed of 65 handwritten scores, from 6 different authors. All the scores in the dataset were reduced to gray-level information. An average value for the best possible global threshold for each image was obtained using five different people. A subset of 10 scores was manually segmented to be used as ground truth for the evaluation procedure.[8] For global thresholding processes, the authors chose three different measures: difference from reference threshold (DRT); misclassification error (ME); and comparison between results of staff finder algorithms applied to each binarized image. For the adaptive binarization, two new error rates were included: the missed object pixel rate and the false object pixel, dealing with loss in object pixels and excess noise, respectively.

Three datasets are accessible to evaluate the algorithms for staff line detection and removal: the Synthetic Score

---

[7] http://www.inescporto.pt/~jsc/ReproducibleResearch.html.

[8] The process to create ground-truths is to binarize images by hand, cleaning all the noise and background, making sure nothing more than the objects remains. This process is extremely time-consuming and for this reason only 10 scores were chosen from the entire dataset.

Database by Christoph Dalitz,[9] the CVC-MUSCIMA Database by Alicia FornTs[10] and the Handwritten Score Database by Jaime Cardoso[11][16]. The first consists of 32 ideal music scores where different deformations were applied covering a wide range of music types and music fonts. The deformations and the ground truth information for these syntactic images are accessible through the MusicStaves toolkit from the Generalized Algorithms and Methods for Enhancement and Restoration of Archives (Gamera) framework.[12] Dalitz et al. [26] put their database available together with their source code. Three error metrics based on individual pixels, staff-segment regions, and staff interruption location were created to measure the performance of the algorithms for staff line removal. The CVC-MUSCIMA Database contains 1,000 music sheets of the same 20 music scores which were written by 50 different musicians, using the same pen and the same kind of music paper with printed staff lines. The images of this database were distorted using the algorithms from Dalitz et al. [26]. In total, the dataset has 12000 images with ground truth for the staff removal task and for writer identification. The database created by Cardoso comprises 50 real scores with real positions of the staff lines and music symbols obtained manually.

Two datasets are available to train a classifier for the music symbols recognition step. Desaedeleer [30],[13] in his open source project to perform OMR, has created 15 classes of printed music symbols with a total of 725 objects. Rebelo et al. [83] have created a dataset with 14 classes of printed and handwritten music symbols, each of them with 2,521 and 3,222 symbols, respectively.[14]

As already mentioned in this article, there are several commercial OMR systems available[15] and their recognition accuracy, as claimed by the distributor, is about 90% [6,51]. However, this is not a reliable value. It is not specified what music score database were used and how this value was estimated. The measurement and comparison in terms of performance of different OMR algorithms is an issue that has already been widely considered and discussed (e.g. [6,51,61,97]). As referred in [6] the meaning of music recognition depends on the goal in mind. For instance, some applications aim to produce an audio record from a music score through document analysis, while others only want to transcode a score into interchange data formats. These different objectives hinder the creation of a common methodology

to compare the results of an OMR process. Having a way of quantifying the achievement of OMR systems would be truly significant for the scientific community. On the one hand, by knowing the OMR's accuracy rate, we can predict production costs and make decisions on the whole recognition process. On the other hand, quantitative measures bring progress to the OMR field, thus making it a reference for researchers [6].

Jones et al. [51] address several important shortcomings to take into consideration when comparing different OMR systems: (1) each available commercial OMR systems has its own output interface —for instance, PhotoScore works with Sibelius, a music notation software—becoming very difficult to assess the performance of the OMR system alone, (2) the input images are not exactly the same, having differences in input format, resolution and image-depth, (3) the input and output have different format representations—for instance .mus format is used in the Finale software and not in Sibelius software, and (4) the differences in the results can be induced by semantic errors. In order to measure the performance of the OMR systems, Jones et al. [51] also suggest an evaluation approach with the following aspects: (1) a standard dataset for OMR or a set of standard terminology is needed to objectively and automatically evaluate the entire music score recognition system, (2) a definition of a set of rules and metrics, encompassing the key points to be considered in the evaluation process, and (3) the definition of different ratios for each kind of error.

Similarly, Bellini et al. [6] proposed two assessment models focused on basic and composite symbols to measure the results of the OMR algorithms. The motivation for these models was the result of opinions from copyists and OMR system builders. The former give much importance to details such as changes in primitive symbols or annotation symbols. The latter, in contrast with the copyists, give more relevance to the capability of the system to recognize the most frequently used objects. The authors defined a set of metrics to count the recognized, missed, and confused music symbols to reach the recognition rate of basic symbols. Furthermore, since a proper identification of a primitive symbol does not mean that its composition is correct, the characterization of the relationships with other symbols is also analyzed. Therefore, a set of metrics has been defined to count categories of recognized, faulty, and missed composite symbols.

Szwoch [97] proposed a strategy to evaluate the result of an OMR process based on the comparison of MusicXML files from the music scores. One of the shortcomings of this methodology is in the output format of the application. Even though MusicXML is becoming more and more a standard music interchange format, it is not yet used in all music recognition software. Another concern is related to the comparison between the MusicXML files. The same score can be correctly represented by different MusicXML codes, making the one-to-one comparison difficult. Miyao and Haralick [61]
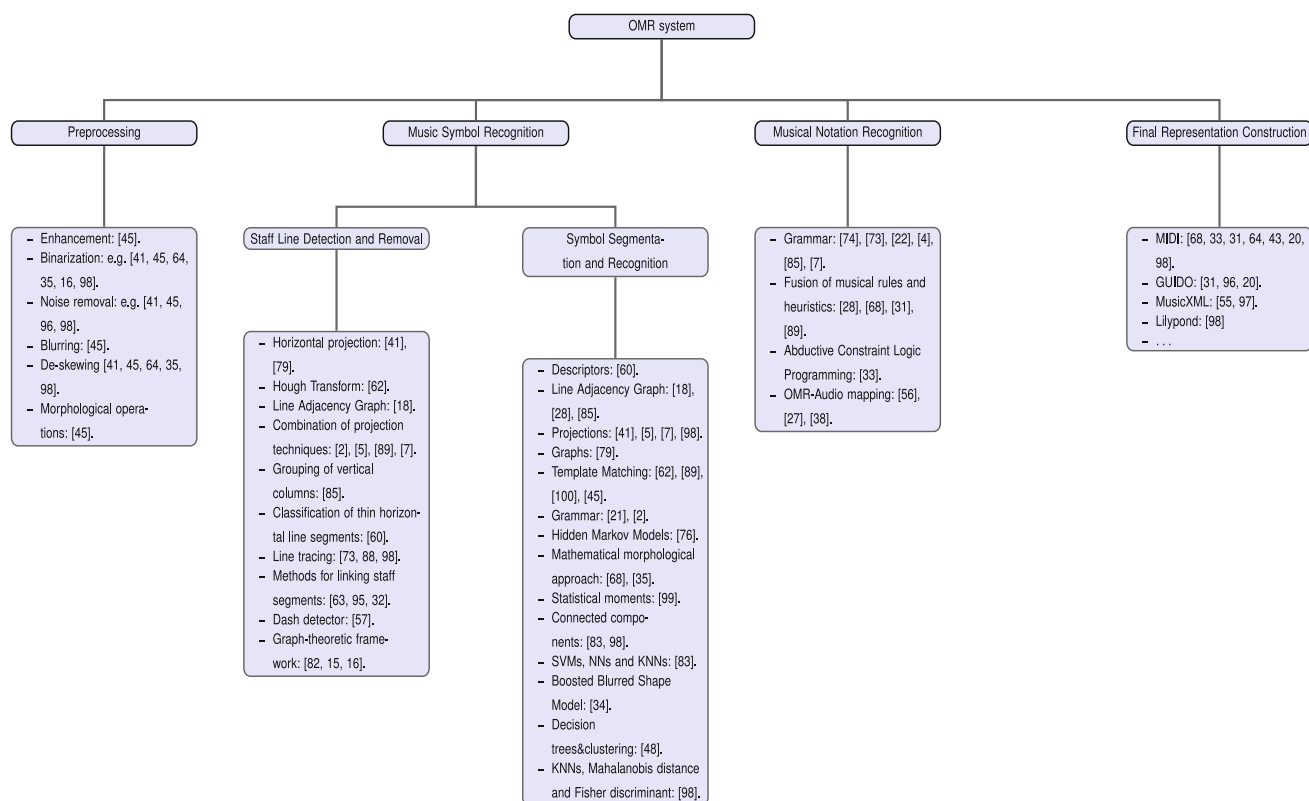

Fig. 8 Summary of the most used techniques in an OMR system

proposed data formats for primitive symbols, music symbols, and hierarchical score representations. The authors stress that when using these specific configurations the researchers can objectively and automatically measure the symbol extraction results and the final music output from an OMR application. Hence, the primitive symbols must include the size and position for each element, the music symbols must have the possible combinations of primitive symbols, and the hierarchical score representation must include the music interpretation. Notwithstanding, this is a difficult approach for comparisons between OMR software since the most of them will not allow the implementation of these models in their algorithms.

7 Open issues and future trends

This paper surveyed several techniques currently available in the OMR field. Figure 8 summarizes the various approaches used in each stage of an OMR system. The most important open issues are related to

- the lack of robust methodologies to recognize handwritten music scores,

- a web-based system providing broad access to a wide corpus of handwritten unpublished music encoded in digital format,
- a master music dataset with different deformations to test different OMR systems,
- and a framework with appropriate metrics to measure the accuracy of different OMR systems.

7.1 Future trends

This present survey unveiled three challenges that should be addressed in future work on OMR as applied to manuscript scores: preprocessing, staff detection and removal, and music symbols segmentation and recognition.

7.1.1 Preprocessing

This is one of the first steps in an OMR system. Hence, it is potentially responsible for generating errors that can propagate to the next steps on the system. Several binarization methods often produce breaks in staff connections, making the detection of staff lines harder. These methods also increase the quantity of noise significantly (see Fig. 6 in Sect. 2). Back-to-front interference, poor paper quality or non-uniform lighting causes these problems. A solution was

proposed in [72] (BLIST). Performing a binarization process using prior knowledge about the content of the document can ensure better results, because this kind of procedure conserves the information that is important to OMR. Moreover, the work proposed in [17] encourages further research in using gray-level images rather than using binary images. Similarly, new possibilities exist for music score segmentation by exploiting the differences in the intensity of gray pixels of the ink and the intensity of gray pixels of the paper.

7.1.2 Staff detection and removal

Some of the state-of-the-art algorithms are capable of performing staff line detection and removal with a good degree of success. Cardoso et al. [16] present a technique to overcome the existing problems in the staff lines of the music scores, by suggesting a graph-theoretic framework (see end of Sect. 3). The promising results promote the utilization and development of this technique for staff line detection and removal in gray-level images. In order to test the various methodologies in this step the authors suggest to the researchers the participation in the staff line removal competition which in 2011 was promoted by International Conference on Document Analysis and Recognition (ICDAR).[16]

7.1.3 Music symbols segmentation and recognition

A musical document has a bidimensional structure in which staff lines are superimposed with several combined symbols organized around the noteheads. This imposes a high level of complexity in the music symbols segmentation which becomes even more challenging in handwritten music scores due to the wider variability of the objects. For printed music documents, a good methodology was proposed in [89]. The algorithm architecture consists in detecting the isolated objects and computing recognition hypotheses for each of the symbols. A final decision about the object is taken based on contextual information and music writing rules. Rebelo et al. [84] are implementing this propitious technique for handwritten music scores. The authors proposed to use the natural existing dependency between music symbols to extract them. For instance, beams connect eighth notes or smaller rhythmic values and accidentals are placed before a notehead and at the same height. As a future trend, they discuss the importance of using global constraints to improve the results in the extraction of symbols. Errors associated with missing symbols, symbol confusion, and falsely detected symbols can be mitigated, e.g., by querying if the detected symbols' durations amount to the value of the time signature on each bar. The inclusion of prior knowledge of syntactic and semantic musical rules may help the extraction of the handwritten

music symbols and consequently it can also lead to better results in the future.

An important issue that could also be addressed is the role of the user in the process. An automatic OMR system capable of recognizing handwritten music scores with high robustness and precision seems to be a goal difficult to achieve. Hence, interactive OMR systems may be a realistic and practical solution to this problem. MacMillan et al. [59] adopted a learning-based approach in which the process improves its results through experts users. The same idea of active learning was suggested by Fujinaga [40] in which a method to learn new music symbols and handwritten music notations based on the combination of a k-nearest neighbor classifier with a genetic algorithm was proposed.

An interesting application of OMR concerns to online recognition, which allows an automatic conversion of text as it is written on a special digital device. Taking into consideration the current proliferation of small electronic devices with increasing computation power, such as tablets and smartphones, it may be usefully the exploration of such features applied on OMR. Besides, composers prefer the creativity which can only be entirely achieved without restrictions. This implies total freedom of use, not only of OMR softwares, but also paper where they can write their music. Hence, algorithms for OMR are still necessary.

8 Conclusion

Over the past decades, substantial research was done in the development of systems that are able to optically recognize and understand musical scores. An overview through the number of articles produced during the past 40 years in the field of optical music recognition makes us aware of the clear increase in research in this area. The progress in the field spans many areas of computer science: from image processing to graphic representation; from pattern recognition to knowledge representation; from probabilistic encodings to error detection and correction. OMR is thus an important and complex field where knowledge from several fields intersects.

An effective and robust OMR system for printed and handwritten music scores can provide several advantages to the scientific community: (1) an automated and time-saving input method to transform paper-based music scores into a machine-readable symbolic format for several music softwares, (2) enable translations, for instance to Braille notations, (3) better access to music, (4) new functionalities and capabilities with interactive multimedia technologies, for instance association of scores and video excerpts, (5) playback, musical analysis, reprinting, editing, and digital archiving, and (6) preservation of cultural heritage [51].

---

[16] http://www.cvc.uab.es/cvcmuscima/competition/.

In this article, we presented a survey on several techniques employed in OMR that can be used in <u>processing printed and handwritten music scores</u>. We also presented an evaluation of current state-of-the-art algorithms as applied to handwritten scores. The challenges faced by OMR systems dedicated to handwritten music scores were identified, and some approaches for improving such systems were presented and suggested for future development. This survey thus aims to be a reference scheme for any researcher wanting to compare new OMR algorithms against well-known ones and provide guidelines for further development in the field.

References

1. Andronico A, Ciampa A (1995) On automatic pattern recognition and acquisition of printed music, 1982. In: Coüasnon B, Camillerapp J (eds) A way to separate knowledge from program in structured document analysis: application to optical music recognition. International conference on document analysis and recognition, pp 1092–1097

2. Bainbridge D (1997) An extensible optical music recognition system. In: Proceedings of the nineteenth Australasian computer science conference, pp 308–317

3. Bainbridge D, Bell T (2001) The challenge of optical music recognition. Comput Hum 35(2):95–121

4. Bainbridge D, Bell T (2003) A music notation construction engine for optical music recognition. Softw Pract Exp 33(2):173–200

5. Bellini P, Bruno I, Nesi P (2001) Optical music sheet segmentation. In: Proceedings of the first international conference on web delivering of music, pp 183–190

6. Bellini P, Bruno I, Nesi P (2007) Assessing optical music recognition tools. Comput Music J 31:68–93

7. Bellini P, Bruno I, Nesi P (2008) Optical music recognition: architecture and algorithms. In: Interactive multimedia music technologies. IGI Global, Hershey, pp 80–110

8. Bernsen J (2005) Dynamic thresholding of grey-level images, 1986. In: Bieniecki W, Grabowski S (eds) Multi-pass approach to adaptive thresholding based image segmentation. In: Proceedings of the 8th international IEEE conference CADSM

9. 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, Berlin, pp 405–434

10. Brink AD, Pendock NE (1996) Minimum cross-entropy threshold selection. Pattern Recognit 29(1):179–188

11. Burgoyne JA, Ouyang Y, Himmelman T, Devaney J, Pugin L, Fujinaga I (2009) Lyric extraction and recognition on digital images of early music sources. In: Proceedings of the 10th International Society for Music, information retrieval, pp 723–727

12. Burgoyne JA, Pugin L, Eustace G, Fujinaga I (2007) A comparative survey of image binarisation algorithms for optical recognition on degraded musical sources. In: Proceedings of the 8th International Society for Music, information retrieval, pp 509–512

13. Byrd D, Schindele M (2006) Prospects for improving OMR with multiple recognizers. In: Proceedings of the 7th International Society for Music, information retrieval, pp 41–47

14. Capela A, Cardoso JS, Rebelo A, Guedes C (2008) Integrated recognition system for music scores. In: Proceedings of the international computer music conference

15. Cardoso JS, Capela A, Rebelo A, Guedes C (2008) A connected path approach for staff detection on a music score. In: Proceedings of the 15th IEEE international conference on image processing, pp 1005–1008

16. Cardoso JS, Capela A, Rebelo A, Guedes C, Pinto da Costa JF (2009) Staff detection with stable paths. IEEE Trans Pattern Anal Mach Intell 31(6):1134–1139

17. Cardoso JS, Rebelo A (2010) Robust staffline thickness and distance estimation in binary and gray-level music scores. In: Proceedings of The twentieth international conference on pattern recognition, pp 1856–1859

18. Carter NP (1992) Automatic recognition of printed music in the context of electronic publishing, 1989. A critical survey of music image analysis. In: Blostein D, Baird H (eds) Structured document image analysis. Springer, Heidelberg, pp 405–434

19. Chen Q, Sun Q, Heng P, Xia D (2008) A double-threshold image binarization method based on edge detector. Pattern Recognit 41(4):1254–1267

20. Choudhury G, Droetboom M, DiLauro T, Fujinaga I, Harrington B (2000) Optical music recognition system within a large-scale digitization project. In: Proceedings of the International Society for Music information retrieval

21. Coüasnon B (1996) Segmentation et reconnaissance de documents guidTes par la connaissance a priori: application aux partitions musicales. PhD thesis, Universit de Rennes

22. Coüasnon B, Brisset P, Stephan I (1995) Using logic programming languages for optical music recognition. In: Proceedings of the third international conference on the practical application of prolog, pp 115–134

23. Coüasnon B, Camillerapp J (1993) Using grammars to segment and recognize music scores. In: Proceedings of DAS-94: international association for pattern recognition workshop on document analysis systems, pp 15–27

24. Coüasnon B, Camillerapp J (1995) A way to separate knowledge from program in structured document analysis: application to optical music recognition. In: Proceedings of the third international conference on document analysis and recognition, pp 1092–1097

25. Dalitz C (2009) Reject options and confidence measures for knn classifiers. In: Dalitz C (ed) Schriftenreihe des Fachbereichs Elektrotechnik und Informatik Hochschule Niederrhein, vol 8. Shaker Verlag, Maastricht, pp 16–38

26. Dalitz C, Droettboom M, Czerwinski B, Fujigana I (2008) A comparative study of staff removal algorithms. IEEE Trans Pattern Anal Mach Intell 30:753–766

27. Damm D, Fremerey C, Kurth F, Müller M, Clausen M (2008) Multimodal presentation and browsing of music. In: Proceedings of the 10th international conference on multimodal interfaces. ACM, pp 205–208

28. Dan L (1996) Final year project report automatic optical music recognition, technical report

29. de Albuquerque MP, Esquef IA, Gesualdi Mello AR (2004) Image thresholding using tsallis entropy. Pattern Recognit Lett 25(9):1059–1065

30. Desaedeleer AF (2006) Reading sheet music. Master's thesis, Imperial College London, Technology and Medicine, University of London

31. Droettboom M, Fujinaga I, MacMillan K (2002) Optical music interpretation. In: Proceedings of the joint IAPR international workshop on structural, syntactic, and statistical pattern recognition. Springer, Berlin, pp 378–386

32. Dutta A, Pal U, Fornés A, Lladós J (2010) An efficient staff removal approach from printed musical documents. In: Proceedings of the 20th international conference on pattern recognition. IEEE Computer Society, pp 1965–1968

33. Ferrand M, Leite JA, Cardoso A (1999) Hypothetical reasoning: an application to optical music recognition. In: Proceedings of the Appia-Gulp-Prode'99 joint conference on declarative programming, pp 367–381

34. Fornés A, Escalera S, Lladós J, Sánchez G, Radeva P, Pujol O (2007) Handwritten symbol recognition by a boosted blurred shape model with error correction. In: Proceedings of the 3rd Iberian conference on pattern recognition and image analysis, part I. Springer, Berlin, pp 13–21

35. Fornés A, Sánchez G (2005) Primitive segmentation in old handwritten music scores. In: Liu W, Llads J (eds) Graphics recognition. Ten years review and future perspectives. Lecture notes in computer science, vol 3926. Springer, Berlin, pp 279–290

36. Fornés A, Lladós J, Sánchez G, Bunke H (2008) Writer identification in old handwritten music scores. In: Proceedings of the 2008 the eighth IAPR international workshop on document analysis systems. IEEE Computer Society, pp 347–353

37. Fornés A, Lladós J, Sánchez G, Bunke H (2009) On the use of textural features for writer identification in old handwritten music scores. In: Proceedings of the 2009 10th international conference on document analysis and recognition. IEEE Computer Society, pp 996–1000

38. Fremerey C, Müller M, Kurth F, Clausen M (2008) Automatic mapping of scanned sheet music to audio recordings. In: Proceedings of the 9th International Society for Music, information retrieval, pp 413–418

39. Friel N, Molchanov I (1999) A new thresholding technique based on random sets. Pattern Recognit 32(9):1507–1517

40. Fujinaga I (1996) Exemplar-based learning in adaptive optical music recognition system. In: Proceedings of the international computer music conference, pp 55–60

41. Fujinaga I (2004) Staff detection and removal. In: George S (ed) Visual perception of music notation: on-line and off-line recognition. Idea Group Inc., Hershey, pp 1–39

42. Gatos B, Pratikakis I, Perantonis SJ (2004) An adaptive binarisation technique for low quality historical documents. In: Document analysis systems VI. Lecture notes in computer science, vol 3163. Springer, Berlin, pp 102–113

43. Genfang C, Wenjun Z, Qiuqiu W (2009) Pick-up the musical information from digital musical score based on mathematical morphology and music notation. In: Proceedings of the 2009 first international workshop on education technology and computer science. IEEE Computer Society, pp 1141–1144

44. George S (2004) Lyric recognition and christian music. In: George S (ed) Visual perception of music notation: on-line and off-line recognition. Idea Group Inc., Hershey, pp 198–225

45. Goecke R (2003) Building a system for writer identification on handwritten music scores. In: Proceedings of the IASTED international conference on signal processing, pattern recognition, and applications. Acta Press, Anaheim, pp 205–255

46. Grandvalet Y, Rakotomamonjy A, Keshet J, Canu S (2008) Support vector machines with a reject option. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) Advances in neural information processing systems. MIT Press, Cambridge, pp 537–544

47. Homenda W (2005) Optical music recognition: the case study of pattern recognition. In: Kurzynski M, Puchala E, Wozniak M, zolnierek A (eds) Computer recognition systems. Advances in soft computing, vol 30. Springer, Heidelberg, pp 835–842

48. Homenda W, Luckner M (2006) Automatic knowledge acquisition: recognizing music notation with methods of centroids and classifications trees. In: Proceedings of the international joint conference on neural networks, pp 3382–3388

49. Huang LK, Wang MJJ (1995) Image thresholding by minimizing the measures of fuzziness. Pattern Recognit 28(1):41–51

50. Jain AK, Zongker D (1997) Representation and recognition of handwritten digits using deformable templates. IEEE Trans Pattern Anal Mach Intell 19(12):1386–1391

51. Jones G, Ong B, Bruno I, Ng K (2008) Optical music imaging: music document digitisation, recognition, evaluation, and restoration. In: Interactive multimedia music technologies. IGI Global, Hershey, pp 50–79

52. Kapur J, Sahoo P, Wong A (1985) A new method for gray-level picture thresholding using the entropy of the histogram. Comput Vis Graph Image Process 29(3):273–285

53. Kassler M (2009) Optical character recognition of printed music: a review of two dissertations, 1972. In: Vrist B (ed) Optical music recognition for structural information from high-quality scanned music, technical report

54. Khashman A, Sekeroglu B (2007) A novel thresholding method for text separation and document enhancement. In: Proceedings of the 11th panhellenic conference on informatics

55. Knopke I, Byrd D (2007) Towards musicdiff: a foundation for improved optical music recognition using multiple recognizers. In: Proceedings of the 8th International Society for Music, information retrieval, pp 123–124

56. Kurth F, Müller M, Fremerey C, Chang Y, Clausen M (2007) Automated synchronization of scanned sheet music with audio recordings. In: Proceedings of the 8th International Society for Music, information retrieval, pp 261–266

57. Leplumey I, Camillerapp J, Lorette G (1993) A robust detector for music staves. In: Proceedings of the international conference on document analysis and recognition, pp 902–905

58. Luth N (2002) Automatic identification of music notations. In: Proceedings of the first international symposium on cyber worlds. IEEE Computer Society, pp 203–210

59. MacMillan K, Droettboom M, Fujinaga I (2002) Gamera: optical music recognition in a new shell. In Proceedings of the international computer music conference, pp 482–485

60. Mahoney JV (1992) Automatic analysis of music score images, 1982. A critical survey of music image analysis. In: Blostein D, Baird H (eds) Structured document image analysis. Springer, Heidelberg, pp 405–434

61. Miyao H, Haralick RM (2000) Format of ground truth data used in the evaluation of the results of an optical music recognition system. In: IAPR workshop on document analysis systems, pp 497–506

62. Miyao H, Nakano Y (1996) Note symbol extraction for printed piano scores using neural networks. IEICE Trans Inform Syst E79-D:548–554. ISSN: 0916–8532

63. Miyao H, Okamoto M (2004) Stave extraction for printed music scores using DP matching. J Adv Comput Intell Intell Inform 8:208–215

64. Ng K (2004a) Optical music analysis for printed music score and handwritten manuscript. In: George S (ed) Visual perception of music notation: on-line and off-line recognition. Idea Group Inc., Hershey, pp 1–39

65. Ng K (2004b) Optical music analysis for printed music score and handwritten music manuscript. In: George S (ed) Visual perception of music notation: on-line and off-line recognition. Idea Group Inc., Hershey, pp 108–127

66. Ng K, Boyle R (1996) Recognition and reconstruction of primitives in music scores. Image Vis Comput 14(1):39–46

67. Ng K, Boyle R, Cooper D (1995) Domain knowledge enhancement of optical music score recognition, technical report

68. Ng K, Cooper D, Stefani E, Boyle R, Bailey N (1999) Embracing the composer: optical recognition of handwritten manuscripts. In: Proceedings of the international computer music conference, pp 500–503

69. Niblack W (2003) An introduction to digital image processing, 1986. Comparison of some thresholding algorithms for text/background segmentation in difficult document images. In: Leedham G, Yan C, Takru K , Tan J, Mian L (eds) Proceedings of the seventh international conference on document analysis and recognition, pp 859–864

70. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66

71. Pal N, Pal S (2004) Entropic thresholding, 1989. In: Sezgin M, Sankur B (eds) Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imaging 13(1):146–165

72. Pinto T, Rebelo A, Giraldi G, Cardoso JS (2011) Music score binarization based on domain knowledge. In: Pattern recognition and image analysis. Lecture notes in computer science, vol 6669. Springer, Heidelberg, pp 700–708

73. Prerau D (1992) Computer pattern recognition of standard engraved music notation, 1970. A critical survey of music image analysis. In: Blostein D, Baird H (eds) Structured document image analysis. Springer, Heidelberg, pp 405–434

74. Prerau D (1992) Optical music recognition using projections, 1988. A critical survey of music image analysis. In: Blostein D, Baird H (eds) Structured document image analysis. Springer, Heidelberg, pp 405–434

75. Pruslin D (1992) Automatic recognition of sheet music, 1966. A critical survey of music image analysis. In: Blostein D, Baird H (eds) Structured document image analysis. Springer, Heidelberg, pp 405–434

76. Pugin L (2006) Optical music recognition of early typographic prints using Hidden Markov models. In: Proceedings of the International Society for Music, information retrieval, pp 53–56

77. Pugin L, Burgoyne J, Fujinaga I (2007a) Goal-directed evaluation for the improvement of optical music recognition on early music prints. In: Proceedings of the 7th ACM/IEEE-CS joint conference on digital libraries. ACM, pp 303–304

78. Pugin L, Burgoyne JA, Fujinaga I (2007b) MAP adaptation to improve optical music recognition of early music documents using Hidden Markov models. In: Proceedings of the 8th International Society for Music, information retrieval, pp 513–516

79. Randriamahefa R, Cocquerez JP, Fluhr C, Pepin F, Philipp S (1993) Printed music recognition. In: Proceedings of the second international conference on document analysis and recognition, pp 898–901

80. Read G (1969) Music notation: a manual of modern practice, 2 edn. Taplinger, New York. ISBN: 0-8008-5459-4

81. Rebelo A (2008) New methodologies torwads an automatic optical recognition of handwritten musical scores. Master's thesis, School of Sciences, University of Porto

82. Rebelo A, Capela A, Pinto da Costa JF, Guedes C, Carrapatoso E, Cardoso JS (2007) A shortest path approach for staff line detection. In: Proceedings of the third international conference on automated production of cross media content for multi-channel, distribution, pp 79–85

83. Rebelo A, Capela G, Cardoso JS (2010) Optical recognition of music symbols: a comparative study. Int J Document Anal Recognit 13:19–31

84. Rebelo A, Paszkiewicz F, Guedes C, Marcal A, Cardoso JS (2011) A method for music symbols extraction based on musical rules. In: Bridges: mathematical connections in art, music, and science, pp 81–88

85. Reed KT, Parker JR (1996) Automatic computer recognition of printed music. In: Proceedings of the 13th international conference on pattern recognition, vol 3, pp 803–807

86. Ridler T, Calvard S (1995) Picture thresholding using an iterative selection method, 1978. In: Venkateswarlu N (ed) Implementation

of some image thresholding algorithms on a connection machine-200. Pattern Recognit Lett 16(7):759–768

87. Riley J, Fujinaga I (2003) Recommended best practices for digital image capture of musical scores. OCLC Syst Serv 19(2):62–69

88. Roach JW, Tatem JE (1992) Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment, 1988. A critical survey of music image analysis. In: Blostein D, Baird H (eds) Structured document image analysis. Springer, Heidelberg, pp 405–434

89. Rossant F, Bloch I (2007) Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection. EURASIP J Appl Signal Process 2007(1):160

90. Sahoo P, Wilkins C, Yeager J (1997) Threshold selection using renyi's entropy. Pattern Recognit 30(1):71–84

91. Sezan M (1985) A peak detection algorithm and its application to histogram-based image data reduction. Graph Models Image Process 29:47–59

92. Sezgin M, Sankur B (2004) Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imaging 13(1):146–165

93. Sheridan S, George S (2004) Defacing music score for improved recognition. In: Abraham G, Rubinstein BIP (eds) Proceedings of the second Australian undergraduate students' computing conference. Australian undergraduate students' computing conference, pp 1–7

94. Sousa R, Mora B, Cardoso JS (2009) An ordinal data method for the classification with reject option. In: Proceedings of the eighth international conference on machine learning and applications, pp 746–750

95. Szwoch M (2005) A robust detector for distorted music staves. In: Computer analysis of images and patterns. Lecture notes in computer science. Springer, Berlin, pp 701–708

96. Szwoch M (2007) Guido: a musical score recognition system. In: Proceedings of the ninth international conference on document analysis and recognition, vol 2. IEEE Computer Society, pp 809–813

97. Szwoch M (2008) Using musicxml to evaluate accuracy of omr systems. In: Stapleton G, Howse J, Lee J (eds) Diagrammatic representation and inference. Lecture notes in computer science, vol 5223. Springer, Berlin, pp 419–422

98. Tardón LJ, Sammartino S, Barbancho I, Gómez V, Oliver A (2009) Optical music recognition for scores written in white mensural notation. EURASIP J Image Video Process. Article ID: 843401. ISSN: 1687–5176

99. Taubman G (2005) Musichand: a handwritten music recognition system, technical report

100. Toyama F, Shoji K, Miyamichi J (2006) Symbol recognition of printed piano scores with touching symbols. In: Proceedings of the international conference on pattern recognition. IEEE Computer Society, pp 480–483

101. Trier O, Jain A (1995) Goal-directed evaluation of binarization methods. IEEE Trans Pattern Anal Mach Intell 17(12):1191–1201

102. Trier O, Taxt T (1995) Evaluation of binarization methods for document images. IEEE Trans Pattern Anal Mach Intell 17(3):312–315

103. Tsai D-M (1995) A fast thresholding selection procedure for multimodal and unimodal histograms. Pattern Recognit Lett 16(6):653–666

104. Yanowitz S, Bruckstein A (1989) A new method for image segmentation. Comput Vis Graph Image Process 46(1):82–95