ELSEVIER

Pattern Recognition Letters 23 (2002) 1129–1141

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A global method for music symbol recognition in typeset music sheets

Florence Rossant *

Institut Supérieur d'Electronique de Paris (ISEP), Telecom Signal Image Department, 21, rue d'Assas, 75006 Paris, France

Received 15 November 2000; received in revised form 24 September 2001

Abstract

This paper presents an optical music recognition (OMR) system that can automatically recognize the main musical symbols of a scanned paper-based music score. Two major stages are distinguished: the first one, using low-level pre-processing, detects the isolated objects and outputs some hypotheses about them; the second one has to take the final correct decision, through high-level processing including contextual information and music writing rules. This article exposes both stages of the method: after explaining in detail the first one, the symbol analysis process, it shows through first experiments that its outputs can efficiently be used as inputs for a high-level decision process. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Optical music recognition; Template matching; Set of recognition hypotheses; Global decision under contextual constraints

1. Introduction

Optical music recognition (OMR) is the process of interpreting automatically a scanned music score. It has been an active research topic since 1967 (Carter et al., 1988; Blostein and Baird, 1992) although it seems to be at first sight a rather simple problem of document analysis: reasonable number of symbols, strict location on the staff lines, strong rules of music writing. In fact, it is a much more difficult and computationally expensive task than it seems to be, even in the case of typeset music. According to Blostein and Baird (1992), one major problem results from segmentation difficulties, due to connection of normally separated primitives, or broken objects. Since then researches have tended to integrate as much a priori knowledge as possible, deduced from music writing rules. For instance, Couasnon and Camillerapp (1994) proposed a segmentation and recognition process entirely controlled by a grammar which formalises relative positions between objects. More often, syntactic rules are introduced later after a first recognition has been made, in order to detect and to a certain extent correct some errors, such as note length errors (e.g. Couasnon and Rétif, 1995; Blostein and Haken, 1999). Even if some existing

* Tel.: +33-149545262; fax: +33-149545251.
E-mail address: florence.rossant@isep.fr (F. Rossant).

PII: S0167-8655(02)00036-3

systems go back to image level processing in order to re-label primitives, (e.g. Modayur, 1996; Kato and Inokuchi, 1990), no global review of previous decisions is made and corrections can consequently only be local. For this reason, other authors prefer to view the recognition process as a global optimization problem. They propose generally a model of the document to be interpreted and look for the maximum of a function expressing the likelihood of the interpretation (Kopec et al., 1996; Stückelberg and Doermann, 1999).

The method described in this paper comes from the assertion that the final decision has to be taken at the score sheet level, and not only according to local criteria or specific corrections. For example, a natural has to be at the left side of a note, at the same height, but makes sense only if a sharp or a flat has previously appeared in the bar or in the key-signature; a false length of a bar may come from misreading a note flag, as well as from non-recognition or confusion. Consequently, the method proposed here looks likewise for a global solution but is different from global approaches mentioned above. Indeed symbol recognition is not based on sophisticated document models. Moreover, it is computed in two major sequential stages. In the first one, an analysis process outputs for each isolated component the major hypotheses about its nature and eventually its length, but does not take any definitive decision. During the second stage, the decision process takes into account all these ambiguous assumptions and also some global information, such as key-signature or time signature, and results in a global solution that meets music writing rules. This method is therefore based on a “feedforward” model, as decision is made on hypotheses earlier generated, and, in this sense, is similar to Fahmy and Blostein’s one (1998). It is also noteworthy that feedforward models have been studied in other domains of document analysis (Baird and Ittner, 1994).

This paper begins with an overview of the whole program, and explains the pre-processing that is needed for symbol recognition. After that, it focuses on the recognition process itself. The first stage, the symbol analysis process, is exposed in detail. The first experiments of a global decision and the results obtained are then reported, which suggest promising improvements of the method.

2. General overview

The inputs of the program are the binary image of the score sheet (1 for a black pixel, 0 for a white one), scanned at a resolution of 300 dpi and some global information, the clef, the key-signature, and the time signature. For the moment, the system handles only typeset monophonic music, excluding handwritten music, and allowing only one voice per staff. It makes the off-line recognition of the following symbols: bar lines, rests, notes (note heads and possible flags, to get pitch and value), accidentals and duration dots attributed to notes. It is sufficient to output the melody and all other writing has to be ignored.

The processing flow, illustrated by Fig. 1, can be divided into three main steps, which are exposed in the following three sections. It starts with all the pre-processing that is needed to make the isolation of symbols. Then, the analysis process, mainly based on template matching, outputs a set of hypotheses expressing ambiguities or uncertainty about the label or the length of each detected object. Finally the global decision process evaluates bar by bar all consistent hypothesis grouping and retains the most likely solution matching music rules.

3. Pre-processing

3.1. Staff detection and skew correction

Music has to be read staff by staff. Staff lines contain important information:

- The staff spacing, in pixel units, conveys the scale of the score and gives a fundamental unit used afterwards to normalize distances and sizes.
- The pitch of a note is deduced from the vertical position of its head, relative to the staff lines. Other musical symbols, such as rests, have to be sought around the third staff line.

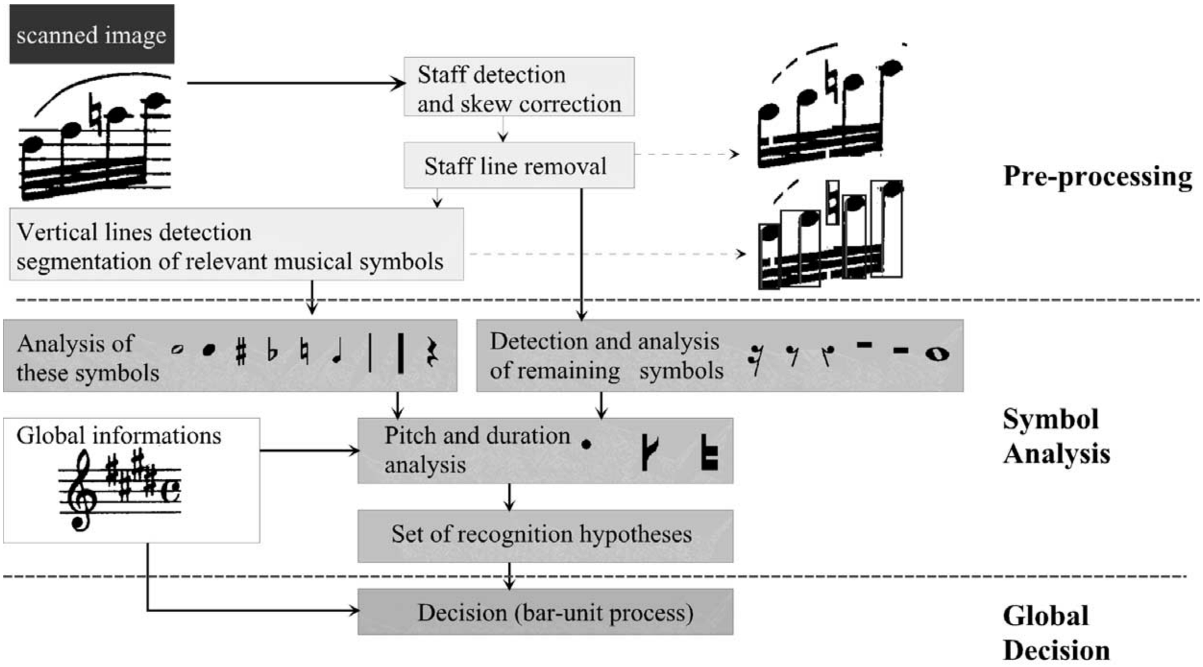

scanned image
Staff detection and skew correction
Staff line removal
Vertical lines detection segmentation of relevant musical symbols
Pre-processing
Analysis of these symbols
Detection and analysis of remaining symbols
Global informations
Pitch and duration analysis
Set of recognition hypotheses
Symbol Analysis
Decision (bar-unit process)
Global Decision

Fig. 1. The processing flow.

- The inclination of staff lines allows us to correct the image skew.

The following process is implemented to detect the image inclination and correct it. The image $i(x,y)$, of W pixels width per H pixels height, is split into two vertical halves. The right half is vertically shifted by h pixels, h being in the range $[-h_o,+h_o]$. If $C(h)$ is the correlation function defined by $C(h)=\sum_{y=1}^{H}\sum_{x=1}^{W/2} i(x,y)\cdot i(x+W/2,y+h)$, the global maximum of this function is obtained for $h_{\max}$ which superimposes the staff lines of both sides of the image. Reported to the half width of the image, $W/2$, this value leads to the inclination angle of the score sheet. When $h_{\max}$ differs from zero, the inclination of the image is simply corrected by the following transform of each pixel: $i'(x,y)=i(x,y+2\cdot h_{\max}x/W)$. This supposes that careful scanning is made so that image skew is less than one degree and so that rotation of a typical symbol can be neglected. This simple process has proven to be sufficient in our experiments.

After that, a horizontal projection profile is conventionally computed to pinpoint the staff lines (Baumann and Dengel, 1992; Marinai and Nesi, 1999), and get the average staff spacing.

3.2. Symbol segmentation

Staff lines to some extent disturb recognition, because they connect and intersect most of the musical symbols. That is why the segmentation algorithm begins with the removal of all staff line segments that do not overlap with a symbol. Each staff line, located at the y_0 position, is horizontally tracked. For each x coordinate, the length of the vertical segment above and under y_0 is computed. If both of these measures are less than the typical thickness of a staff line (the quarter of staff spacing), the segment is removed; otherwise, it is kept. Fig. 2 illustrates the results of this process.

The second stage of the segmentation process concerns all the musical symbols which are featured by a vertical segment longer than 1.5 staff spacing: bar lines, notes (whole notes excepted), flats, sharps, naturals, and some crotchet rests (Fig. 4). In a first step, the longest vertical segment, longer than 1.5 staff spacing, is detected in each

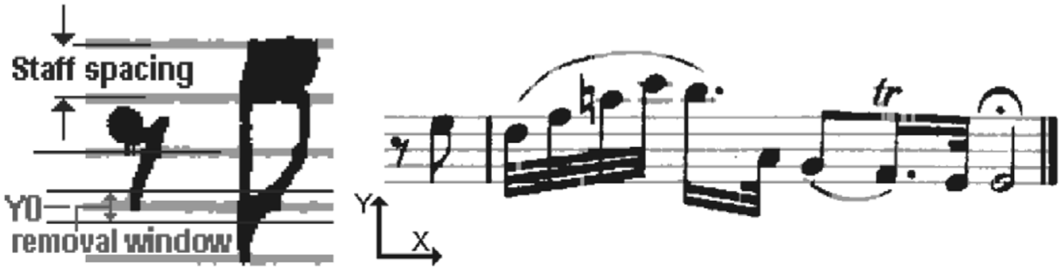


Fig. 2. Staff lines removal (removed pixels in light gray).

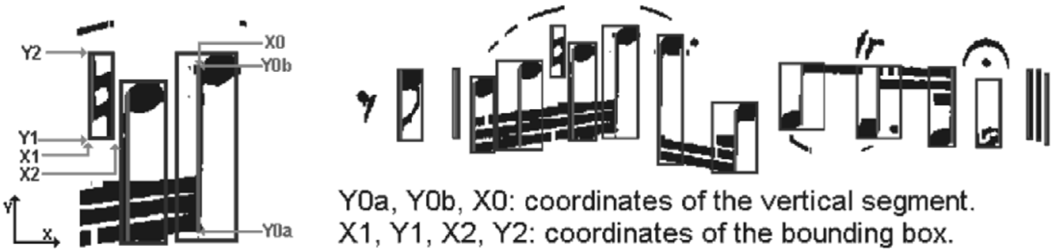

Y2
Y1
X1
X2
X0
Y0b
Y0a
Y
X
tr
Y0a, Y0b, X0: coordinates of the vertical segment.
X1, Y1, X2, Y2: coordinates of the bounding box.

Fig. 3. Isolation of musical symbols featured by a vertical segment.

column of the staff area. These segments may belong to a vertical line or be part of a thick horizontal line, such as noisy interconnected beams of note groups. So, in a second step, an analysis window is moved along the x coordinate in order to analyze adjacent segments and keep only the main segment of each vertical line. It is then used as seed of a region growing and image-labelling algorithm, so that the relevant musical symbol is isolated and precisely located by a bounding box, as shown in Fig. 3.

3.3. Discussion

These pre-processing algorithms are simple but do not resolve all difficulties. In the worst case, staff detection may fail when too many symbols aligned in an almost horizontal direction blur horizontal-profile peaks. But this arises very rarely. The algorithm has been tested on about fifty music sheets. In only one case, skew was properly corrected but one staff was not detected.

The staff line removal algorithm may also remove parts of symbols, such as flats or half notes, or beams binding notes together (Fig. 2). On the other hand, some parts of staff lines are not removed when they are slightly warped. These segmentation defects cause recognition ambiguities, that will have to be handled by the global decision process, as explained below in Section 5. Anyway, it would be better for staff line detection and removal to use a more robust method, such as Carter and Bacon's one (1992), based on line-adjacency graph.

More dramatic is the non-detection of a thin vertical broken line. Indeed, a segment may be rejected, and consequently the relevant symbol definitively lost. Poulain d'Andecy et al. (1994) has already dealt with this subject through Kalman filtering.

4. Symbol analysis

4.1. Template matching

Symbol analysis is mainly based on template matching. A set of models is used to compute the correlation with the symbol that has to be identified. Let us define the correlation between the model k, M^k, of size $d_x^k \cdot d_y^k$, and the tested shape, at the (x,y) position in the image I:

$$C^k(x,y) = \frac{1}{d_x^k \cdot d_y^k} \sum_{(i,j)\in M^k} M^k(i,j) \cdot I'(x+i, y+j),$$

$$M^k(i,j) = \begin{cases} -1 & \text{for a white pixel,} \\ 1 & \text{for a black pixel,} \end{cases}$$

$$I'(i,j) = \begin{cases} -1 & \text{if } I(i,j) = 0, \\ 1 & \text{if } I(i,j) = 1. \end{cases} \tag{1}$$

In the case of perfect superimposition between shape and model, the result will reach the maximum score of 1.0. The score decreases with the number of pixels that differ from the model. In template matching, the correlation is computed for several (x,y) coordinates. So, the highest score obtained in (x_k, y_k), the only interesting one, is a measure of similarity and of localization.

Various methods have been used in OMR to recognize symbols: morphological techniques (Modayur, 1996), neural networks (Miyao and Nakano, 1995), use of geometrical features and moments (Fujinaga et al., 1991). Template matching is rarely mentioned in the literature as a possible recognition technique of musical primitives, but it is particularly adapted to the proposed method for the following reasons:

- As outlined in Section 3.3, this method is not based on sophisticated segmentation algorithms and there may be some errors, due to the connection of normally disconnected objects, over-segmentation of symbols, or imperfect staff line removal. But even in these cases, the correlation score can be computed and remains significant, if the presence of the object is correctly identified. The detection of vertical segments, featuring most musical symbols, and the use of graphical a priori knowledge for the others are largely sufficient.
- We can see the correlation score between an unknown object and a proposed model, representing a symbol class, as a likelihood measure for the underlying classification hypothesis. This measure will be efficiently used in the global decision process.
- As in other recognition methods, and maybe sometimes more dramatically, template matching is sensitive to typewriting variations. But the set of models can be easily adjusted to the analyzed score sheet. This can be made manually, because models are independent of the program, but also automatically as in the proposed method. This process is explained in Section 4.2.
- It is simple to deal with different sizes of printing by storing corresponding bases of models. Staff spacing indicates which base has to be used to analyse the current music sheet. This has not yet been implemented in the program.

Graphical knowledge is used to improve correlation computing. That is why the analysis of symbols featured by a vertical segment is distinguished from the analysis of all others.

4.2. Analysis of symbols featured by a vertical segment

For these symbols, it is interesting to examine the geometrical features extracted through the segmentation process. For example, a component with a narrow bounding box located between the first and the fifth staff lines may be a bar line. Symbols are grouped in four main classes: notes (whole notes excepted), bar lines, accidentals (flat, sharp, natural, grace note), rests (crotchet rest). For each group, a set of five criteria is computed from geometrical features of the bounding box and of the vertical segment, and also from their position relative to the staff lines. When three of them are satisfied, the correlation with the models of the group (Fig. 4) is computed in an area deduced from the bounding box and from the position of the vertical segment. For each tested model k, the correlation score, $C^k(x_k, y_k)$, and the corresponding location, (x_k, y_k), are stored. Three threshold values have been experimentally defined: a *decision threshold* t_d, validating the recognition of a template, a *minimum threshold* t_m, which has always to be reached to store a template as possible, and an *ambiguity threshold* t_a, to deal with a second highest score close to the first one. The model with the highest score $C^{k1}(x_{k1}, y_{k1})$ is stored in memory if

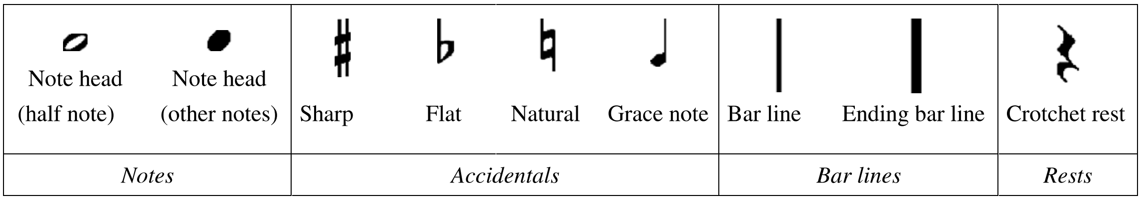


Fig. 4. A set of models featured by a vertical segment.

greater than t_m, but also the hypothesis of absence of a musical symbol if $C^{k1}(x_{k1}, y_{k1})$ is under the decision threshold t_d. In case of ambiguity $(C^{k2}(x_{k2}, y_{k2}) > t_m$ and $(C^{k1}(x_{k1}, y_{k1}) - C^{k2}(x_{k2}, y_{k2})) < t_a)$, the second highest score $C^{k2}(x_{k2}, y_{k2})$ is also stored.

To deal with different fonts, the program tests, at the beginning of the analysis process, several models for each class. After enough processing, it determines which model get the highest correlation score, and stops testing the other ones. This automatic model selection makes the process adaptable to several types of sheets and moreover user-free.

The pitch of possible notes or accidentals is deduced from the y coordinate. The length of notes is obtained from an analysis of the number of possible flags or beams grouping notes, which have to be found at the opposite end of the note stem, and from the correlation computed with a possible duration dot, that may only appear after a note head. Fig. 5 shows an example of this process.

Thus segmentation of symbols featured by a vertical segment is a very important step. Indeed, the extracted geometrical features increase the reliability and the speed of the analysis, by determining a small search area and eliminating impossible classes.

4.3. Analysis of remaining symbols

Musical symbols with no vertical segment, have now to be analyzed (Fig. 6). This step mainly concerns rests, which are sought using a correlation computed around the third staff line, along the free spaces between the bounding boxes. For whole notes, the program has of course to compute the correlation on a larger vertical area. In case of good matching between a model and a symbol, the correlation function presents a maximum peak exceeding the minimum threshold, and this result is stored. As in the previous analysis stage, there can be some ambiguity, when several models get a great correlation score at almost the

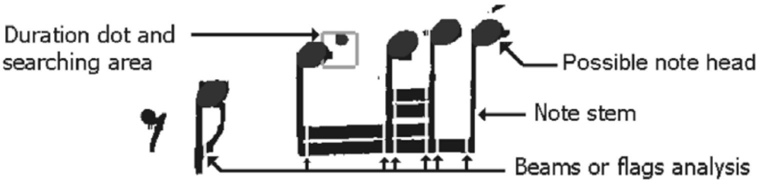


Fig. 5. An example of note analysis.

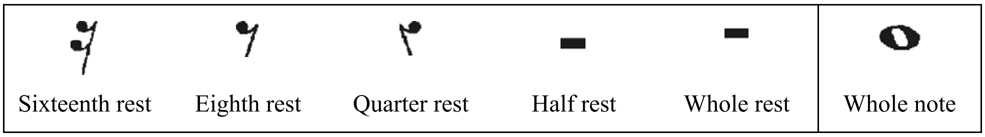


Fig. 6. A set of models used in the analysis of symbols with no vertical segment.

same position. So, in a second step, an iterative process looks, in decreasing order, for the model with the highest score and examines its neighbourhood: when a second match overlaps the first one, it is also stored as second hypothesis. The process runs until every hypothesis is put in the list of possible symbols, as first or second hypothesis.

4.4. Hypothesis merging

All of the analyzed symbols are re-arranged along the horizontal (time) axis. Two pre-processing steps are performed before decision in order to enhance the results. These are briefly described below.

4.4.1. Accidental removing

When an accidental does not match in vertical position the pitch of the following note, it is simply removed from the hypothesis list.

4.4.2. Length analysis

Time signature indicates the whole length of a bar, in terms of beat unit, and the number of beats in the bar. For example, 3/4 implies that the length of the beat is the crotchet, and that there are 3 beats in a bar. As shown in Fig. 7, beams bind together notes into discrete groups, generally a whole number of beats or half beats, so that the beat structure is better isolated. These conventions are used to correct misread duration information. An analysis based on a region growing algorithm extracts the beamed notes, and compares their internal rhythmical organization with the possible common ones, according to the time signature. A maximum of two hypotheses can be made on a note, increasing or decreasing the total length of the group, and changing the smallest possible number of values. Other notes are now supposed to be isolated ones and the number of flags is precisely recounted under this assumption. If the result differs from the previous one, another hypothesis is generated.

At that time, a set of recognition hypotheses is available to make the final decision described below.

5. Global decision

Of course, this stage is crucial and has to be itself the purpose of a deep study. The process described below is only a first experiment in order to confirm that the outputs of the analysis process can be exploited.

To include global information such as time signature and decrease the computing cost, a bar-unit process is implemented to make the final decision. As inputs, we have a set of symbol hypotheses, with their correlation score and their length, and an important music writing rule that has to be respected inside the bar: its whole length, indicated by the time signature.

We can now take a global decision inside each bar. There are five possible levels of assumptions.

L1 – Symbol with the highest correlation score if $t_{\rm d} \leqslant C^{k1}(x_{k1}, y_{k1})$.Or hypothesis of no musical symbol if $t_{\rm m} \leqslant C^{k1}(x_{k1}, y_{k1}) < t_{\rm d}$.
L2 – Symbol with the highest correlation score if $t_{\rm m} \leqslant C^{k1}(x_{k1}, y_{k1}) < t_{\rm d}$.
L3 – Symbol with the second highest score: $t_{\rm m} \leqslant C^{k2}(x_{k2}, y_{k2}), (C^{k1}(x_{k1}, y_{k1}) - C^{k2}(x_{k2}, y_{k2})) < t_{\rm a}$.
L4 – First hypothesis of length change of a note.
L5 – Second hypothesis of length change of a note.

A two-dimensional table is computed: horizontally, the tested objects inside the bar, and vertically the hypotheses made on this object. Duration-dots, of $C^{\rm d\text{-}d}(x_{\rm d\text{-}d}, y_{\rm d\text{-}d})$ correlation score,

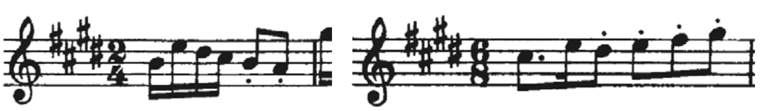
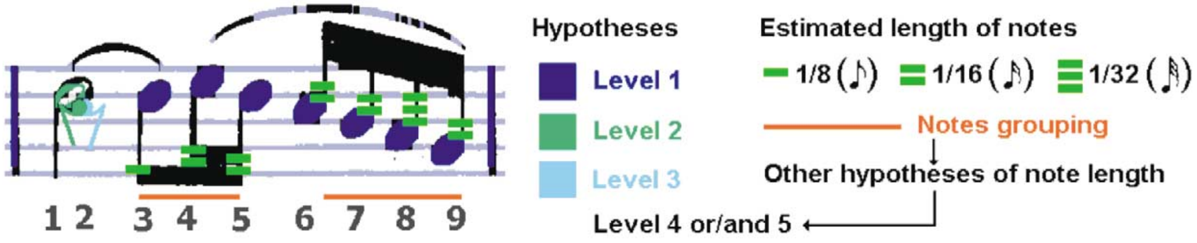


Fig. 7. Two examples of time signature and their common rhythmical grouping.



(a)

	1	2	3	4	5	6	7	8	
L1	NS - -	NS - -	● 1/8 *0.94*	● 1/16 *0.90*	● 1/16 *0.94*	● 1/16 *0.87*	● 1/16 *0.92*	● 1/32 *0.88*	● 1/16 *0.95*
L2	○ 1/2 *0.54*	𝄽 1/4 *0.55*							
L3		𝄾 1/8 *0.51*							
L4								● 1/16 *0.88*	

(b)

Fig. 8. (a) Outputs of analysis process. (b) Two-dimensional table of hypotheses, with symbols, length and correlation scores.

are processed as all other objects and may appear in L1 if $t_d \leqslant C^{d\text{-}d}(x_{d\text{-}d}, y_{d\text{-}d})$ or L2 if $t_m \leqslant C^{d\text{-}d}(x_{d\text{-}d}, y_{d\text{-}d}) < t_d$. The decision algorithm computes the average correlation score of each possible and consistent hypothesis grouping. It chooses the solution of greater score with a minimum number of corrections, giving priority to solutions matching the time signature.

Fig. 8 illustrates an example of ambiguities, and how the correct solution can be found. For symbol 1, the analysis process outputs the assumption of a half note, with a correlation score $t_m \leqslant 0.54 < t_d$. So it is a hypothesis of level L2, the hypothesis of level L1 being no symbol (NS). It is the same for the assumption that symbol 2 is a quarter rest because $t_m \leqslant 0.55 < t_d$. But this time, a hypothesis is also stored in level L3, because the assumption of a eighth rest gets a sufficient correlation score ($t_m \leqslant 0.51$) and is ambiguous ($(0.55 - 0.51) < t_a$). For the remaining symbols, there is only one assumption with correlation score above t_d, so stored in level L1. The group of notes 6–7–8–9 gets a total length of 7/8 of beats. To reach a common rhythmical grouping, the length of note 8 may be changed from a 1/32 note to a 1/16 note. This hypothesis is stored in level L4. Level 5 is not represented because no hypothesis of this kind has been made in this example. Cases in dark grey are also not used. So, there are finally 12 possible groups of hypotheses, and the final correct solution, represented in light grey, validates symbol 1 as half note, eliminates symbol 2, and corrects the length of symbol 8. The total length of the bar-unit is right (4 beats), with an average correlation score of 86.75%.

6. Results and improvements

6.1. Experimental conditions

Experiments have been performed on about fifty music sheets, with all of the previously described algorithms implemented in the program. Care is taken not to train on specific test cases: the examples of the test base come from various publishers, including difficult examples such as thick printing connecting objects; the program has been running without any specific parameter tuning, such as symbol models or decision threshold

Bar lines					**Notes**				
1:Initially right	2:Right corrected	3:Non corrected	4:False corrected	5:Missing or added	1:Initially right	2:Right corrected	3:Non corrected	4:False corrected	5:Missing or added
r1 = 99.5	r2=0.0	r3=0.0	r4=0.0	r5=0.5	r1 = 94.1	r2=3.1	r3=1.6	r4=1.1	r5=0.1
r = 99.5%					**r = 97.2**				

Accidentals					**Rests**				
1:Initially right	2:Right corrected	3:Non corrected	4:False corrected	5:Missing or added	1:Initially right	2:Right corrected	3:Non corrected	4:False corrected	5:Missing or added
r1 = 90.7	r2=2.0	r3=4.2	r4=0.9	r5=2.2	r1 = 75.2	r2=9.5	r3=1.7	r4=2.1	r5=11.5
r = 92.7					**r = 84.7**				

Fig. 9. Recognition rates.

changes. Lastly, scanning has been performed on two different materials, without choosing any specific threshold to get the binary images.

6.2. Results

Typical run-time is around 2 min on a pentium 500 MHz, without any correlation computing optimization or other implementation optimization. Fig. 9 summarizes recognition results for each main kind of symbols, bars, accidentals, notes, rests, without taking into account the exceptional case where a staff was not detected. A symbol is counted as initially correct when the decision process is right to choose the assumption of level L1 (column 1), and as a rightly corrected symbol (column 2) when the decision process is right to choose an assumption of level L2–L5. In the other cases, it is an error: false initial decision not corrected (column 3), right initial decision not kept or non-satisfactory correction (column 4), symbols missing or added (column 5). Five corresponding rates are computed by $ri = ni/\sum_{j=1}^{5} nj$, where ni is the number of symbols belonging to column i. $r = (n1 + n2)/\sum_{j=1}^{5}(nj)$ represents the average recognition rate.

6.3. Discussion

All classes merged, the average recognition rate is over 97%. It is important to note that the rate of wrongly corrected decisions ($r4$) is much lower than the rate of good corrections ($r2$). But average recognition rate may vary from one sheet to another. For clearly printed music scores, the recognition can be perfect and the first hypothesis is generally the right decision. For more difficult scores, strong ambiguities, generally resulting from bad symbol writing, appear more frequently. Fig. 10 shows some common cases of errors: confusion of an accidental with a note, or with another class of accidental, adding of symbols due to false vertical segment detection, misunderstanding of dots.

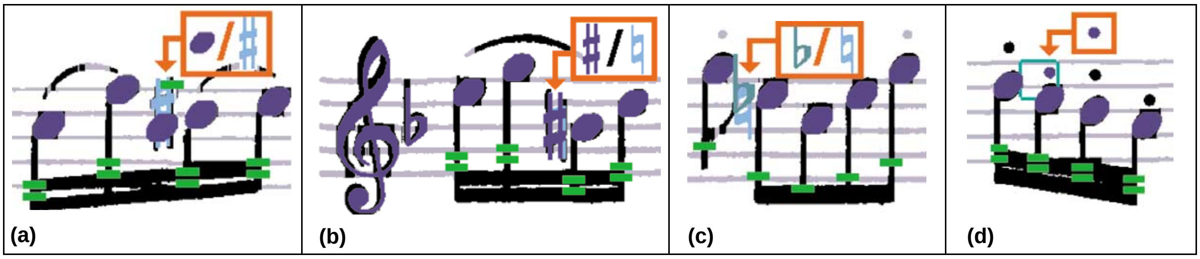


Fig. 10. Some common cases of errors.

The main weakness of the decision algorithm is that it takes its decision using the length of the bar unit as its only constraint. This has some bad consequences, that explain unsatisfactory results for accidentals and especially rests:

- When a bar line is added or not detected, the correct solution can not be found, and in the worst case, this introduces errors. Using a more powerful algorithm for vertical line detection, even broken, would improve bar detection and enhance average results.
- When a bar does not have to match the time signature, for example in cases of pick-up measures that are not yet handled by the program, the length constraint introduces non-existing symbols, most of the time non-existing rests.
- The algorithm is able to correct confusions like that shown in Fig. 10(a). But no arbitration between accidental ambiguities (Fig. 10(b)) nor detection of additional accidentals (Fig. 10(c)) can be made.
- Some false interpretation, for example misunderstanding of dots (Fig. 10(d)), can introduce errors, for example the choice of a shorter rest to compensate the increase of the note length.

6.4. Improvements

We can notice that the correct solution is rarely missing in the outputs of the analysis process. So the effectiveness of the decision process has to be improved. At present, studies on this are under way by adding much more global structural and contextual knowledge, particularly on graphical relative positions, and rules about accidentals. Use of fuzzy models seems to be an interesting approach because it allows to merge such different kinds of knowledge. Experiments have already been made to improve the recognition of accidentals, by expressing a position compatibility degree between an accidental and the following note and a syntactic compatibility degree between the tonality and the accidentals inside a same bar unit (Rossant and Bloch, 2001). Average recognition rate of accidentals has thus been improved by 4.5%. The method is at present being extended to dots and note groupings and leads to promising results.

When the correct solution does not exist in the outputs of the analysis process, it is generally due to serious segmentation problems. As previously commented, it would be interesting to replace the pre-processing algorithms by more robust solutions proposed by other researchers.

6.5. Comparison with commercial software

Although advertisements for commercial OMR packages claim very good recognition rates, users of such systems agree that they are still too error-prone to be of much practical use. Instead of verifying and correcting all the generated results, it is generally faster to convert a score to a computer file by playing the piece on a keyboard connected to a PC and even more convenient to use a music editor. Hence it is interesting to verify if the exposed method can improve this situation.

One of the most recent well-known software for Windows (SmartScore), freely available for demos on the web, has been tested. It is obvious that this program is not able to find a correct solution matching elementary music writing rules, such as length of a bar, whereas the method we propose uses this rule to produce a correct global solution. This is demonstrated by Fig. 11: Fig. 11(a) is an original part of a scanned music sheet, Fig. 11(b) is the score re-edited by the commercial software, Fig. 11(c) shows recognized symbols superimposed on the original image by our program. Regardless of pitch, accidental or grace note errors, 17 of the 45 bars get a wrong length with the commercial software, for four main reasons: false length of a note, symbol missing, symbol adding, confusion. On the contrary, our program reaches much more satisfactory results. Indeed it uses correlation score as confidence measure to arbitrate between several recognition possibilities under the constraint of the whole length of a bar. That is why it is able to reintroduce symbols (e.g rests in red in Fig. 11(c)), to leave definitively others (no adding of symbols in Fig. 11(c)), to deal with ambiguities (e.g. the confusion between note and crotchet, Fig. 11(b) staff 2, would be corrected), and to correct note length errors (e.g Fig. 11(c) staff 2). Moreover, it would be easy to indicate to the user where errors

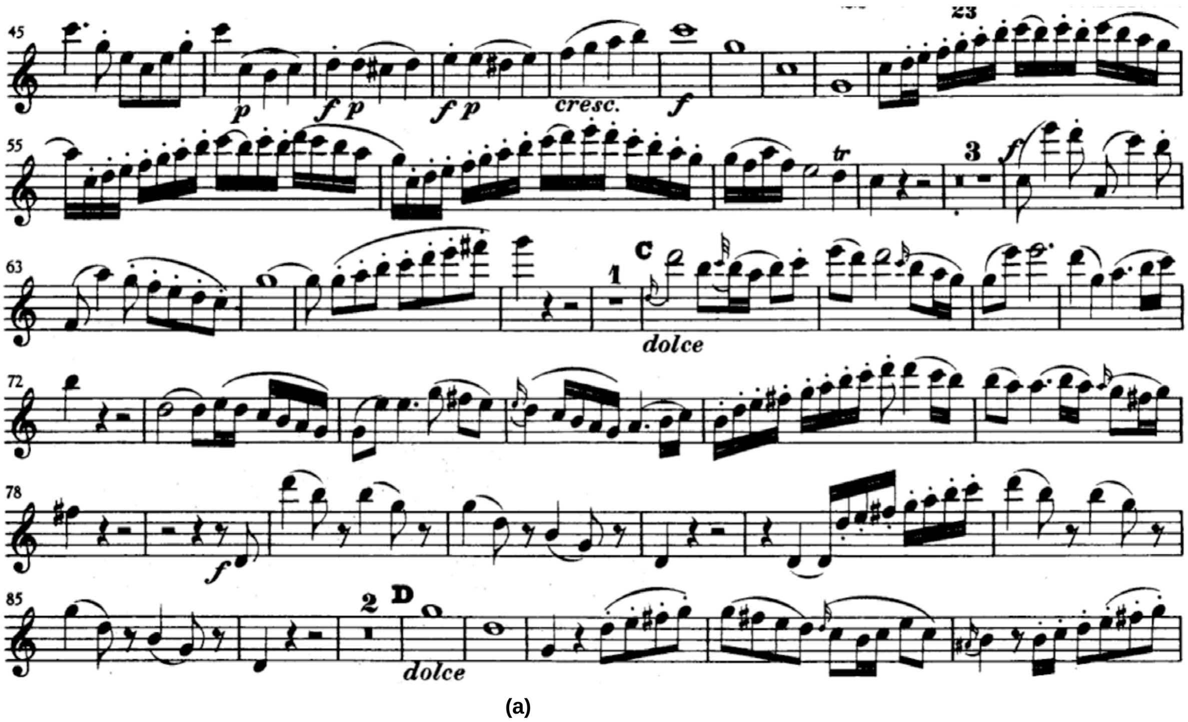


Fig. 11. (a) Original music sheet. (b) Re-edited by commercial software. (c) Results of proposed program.

might be found whenever the program does not provide a satisfactory solution.

6.6. Extension to polyphonic music

Density of information is increased in the case of polyphonic music: more than one symbol may be vertically found. We can now review each stage of the algorithm and examine how it can be extended to polyphonic music.

No specific assumptions have been used during pre-processing. But we can guess that staff line detection and consequently skew removal algorithms will not work as well as they do with monophonic scores, because the increase of symbol density blurs horizontal line detection. So, again, it would be interesting to use a more robust algorithm from the literature.

Symbol analysis can still be based on correlation computing, but symbols will have to be sought in larger areas. For example, several note heads may be found along a vertical segment, and rests are no longer put on the third staff line only.

New music writing rules on graphical relative positions and note groupings will be required to re-arrange symbols in several voices, to define the set of hypotheses and to evaluate them through the decision process. Again, fuzzy logic and possibility theory may be interesting to fusion analysis process outputs with music syntactic rules, taking into account the uncertainty of extracted information, their mutual interactions and relative importance.

7. Conclusion

A global system for OMR has been presented in this article. After staff line detection, skew correction, and segmentation, the recognition process proceeds in two sequential stages to find a global solution. The first one outputs a set of recognition

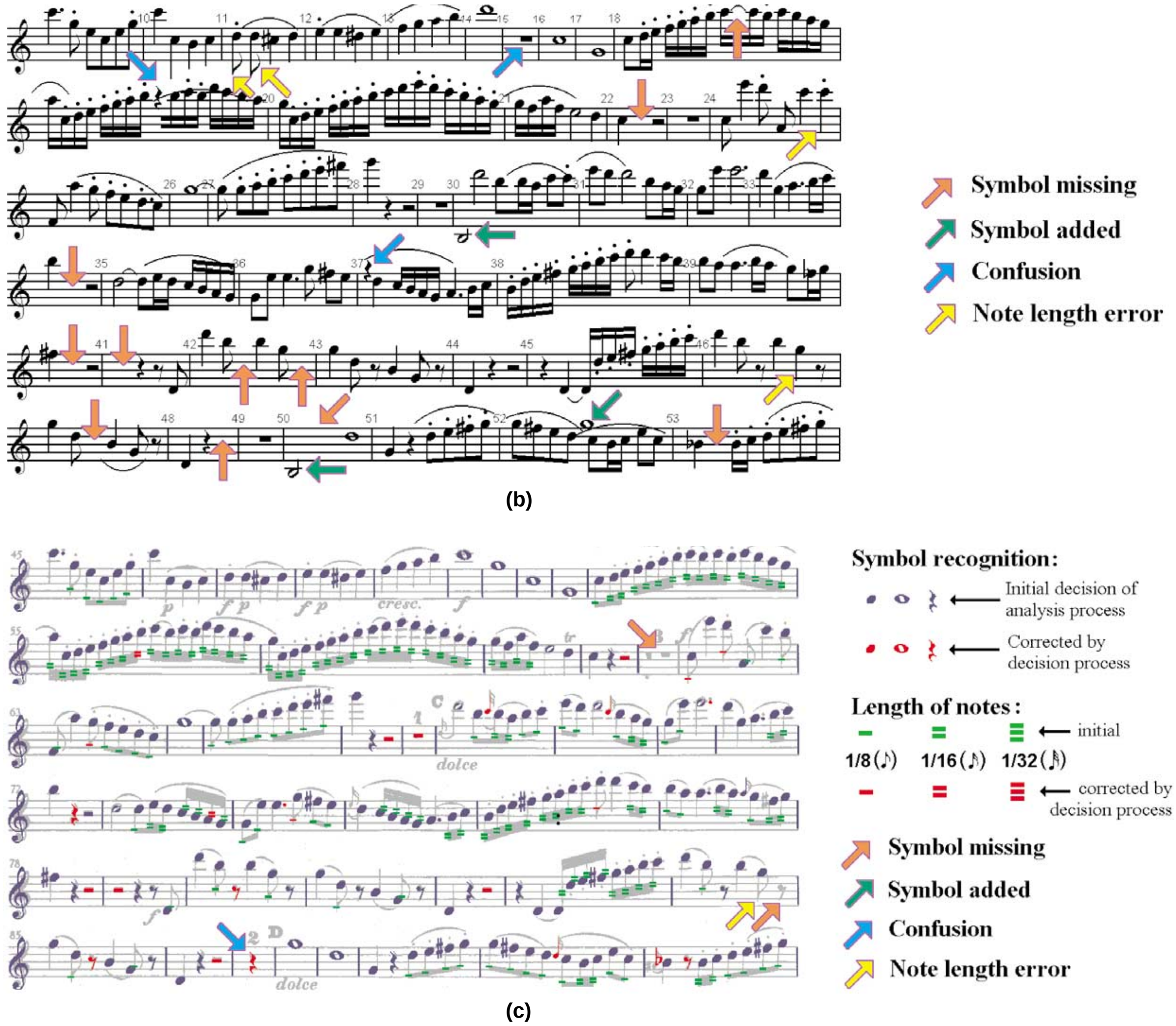


Fig. 11 *(Continued)*

hypotheses, used by the second one to take a global decision matching music writing constraints. First experiments show the efficiency of this methodology. But the high-level global decision process has to be improved by including more contextual knowledge. This is currently being studied.

References

Baird, H., Ittner, D., 1994. Data structures for page readers. In: Proc. IAPR Workshop on Document Analysis Systems, Kaiserlautern, Germany, pp. 323–334.

Baumann, S., Dengel, A., 1992. Transforming printed piano music into midi. In: Proc. IAPR Workshop on SSPR, Bern, Switzerland.

Blostein, D., Baird, H., 1992. A critical survey of music image analysis. In: Baird, H.S., Bunke, H., Yamamoto, K. (Eds.), Structured Document Image Analysis. Springer, Berlin, pp. 405–434.

Blostein, D., Haken, L., 1999. Using diagram generation software to improve diagram recognition: a case study of music notation. IEEE Trans. Pattern Anal. Mach. Intell. 21 (11), 1121–1135.

Carter, N., Bacon, R., 1992. Automatic recognition of printed music. In: Baird, H.S., Bunke, H., Yamamoto, K. (Eds.), Structured Document Image Analysis. Springer, Berlin, pp. 456–465.

Carter, N., Bacon, R., Messenger, T., 1988. The acquisition, representation and reconstruction of printed music by computer: a review. Comput. Hum. 22, 117–136.

Couasnon, B., Camillerapp, J., 1994. Using grammars to segment and recognize music scores. In: Internat. Assoc. Pattern Recognition Workshop on Document Analysis Systems, Kaiserslautern, Germany, pp. 15–27.

Couasnon, B., Rétif, B., 1995. Using a grammar for a reliable full score recognition system. In: Proc. Internat. Comput. Music Conf., Banff, Canada, pp. 187–194.

Fahmy, H., Blostein, D., 1998. A graph-rewriting paradigm for discrete relaxation: application to sheet-music recognition. Internat. J. Pattern Recognition Artificial Intell. 12 (6), 763–799.

Fujinaga, I., Pennycook, B., Alphonce, B., Hogan, K., 1991. Optical music recognition: progress report. In: Proc. Internat. Comput. Music Conf., Montreal, Canada.

Kato, H., Inokuchi, S., 1990. The recognition system of printed piano using musical knowledge and constraints. In: Proc. IAPR Workshop on Syntactic and Structured Pattern Recognition 1990, Murray Hill, NJ, pp. 231–248.

Kopec, G., Chou, P., Maltz, D., 1996. Markov source model for printed music decoding. J. Electron. Imaging 5 (1), 7–14.

Marinai, S., Nesi, P., 1999. Projection based segmentation of musical sheets. In: Proc. ICDAR 1999, Bangalore, India, pp. 515–518.

Miyao, H., Nakano, Y., 1995. Head and stem ext action from printed music scores using a neural network approach. In: Proc. ICDAR 1995, Montreal, Canada, pp. 1074–1078.

Modayur, B., 1996. Music score recognition – a selective attention approach using mathematical morphology. Technical Report, Electrical Engineering Department, University of Washington, Seattle.

Poulain d'Andecy, V., Camillerapp, J., Leplumey, I., 1994. Kalman filtering for segment detection: application to music scores analysis. In: Proc. ICPR 1994, Jerusalem, Israel, pp. 301–305.

Rossant, F., Bloch, I., 2001. Reconnaissance de partitions musicales par modélisation floue et intégration de règles musicales. In: Proc. GRETSI 2001, Toulouse, France, vol. 2, pp. 99–102.

SmartScore Demo for Win95/98/NT Version 1.3: http://www.musitek.com/demopage.html.

Stückelberg, M.V., Doermann, D., 1999. On musical score recognition using probabilistic reasoning. In: Proc. ICDAR 1999, Bangalore, India, pp. 115–118.