Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Fuzzy Sets and Systems 141 (2004) 165–201

FUZZY sets and systems

www.elsevier.com/locate/fss

A fuzzy model for optical recognition of musical scores

Florence Rossant[a], Isabelle Bloch[b,*]

[a]*ISEP - Institut Supérieur d'Electronique de Paris, 21, rue d'Assas - 75006 Paris, France*
[b]*ENST-TSI, CNRS URA 820, 46, rue Barrault - 75013 Paris, France*

Received 31 May 2002; received in revised form 19 February 2003; accepted 25 February 2003

Abstract

Optical music recognition aims at reading automatically scanned scores in order to convert them in an electronic format, such as a midi file. We only consider here classical monophonic music: we exclude any music written on several staves, but also any music that contains chords. In order to overcome recognition failures due to the lack of methods dealing with structural information, non-local rules and corrections, we propose a recognition approach integrating structural information in the form of relationships between symbols and of musical rules. Another contribution of this paper is to solve ambiguities by accounting for sources of imprecision and uncertainty, within the fuzzy set and possibility theory framework. We add to a single symbol analysis several rules for checking the consistency of hypotheses: graphical consistency (compatibility between accidental and note, between grace note and note, between note and augmentation dot, etc.), and syntactic consistency (accidentals, tonality, metric). All these rules are combined in order to lead to better decisions. Experimental results on 65 music sheets show that our approach leads to very good results, and is able to correct errors made by other approaches, such as the one of SmartScore.

Keywords: Optical music recognition; Image processing; Pattern recognition; Fuzzy sets and possibility theory; Flexible rules; Fusion; Structural music information

1. Introduction

Optical music recognition aims at reading automatically scanned scores in order to convert them in an electronic format, such as a midi file. We only consider here classical monophonic music: we exclude any music written on several staves, but also any music that contains chords.

* Corresponding author. Tel.: +33-1-45-817585; fax: +33-1-45-813794.
E-mail addresses: florence.rossant@isep.fr (F. Rossant), isabelle.bloch@enst.fr (I. Bloch).

0165-0114/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0165-0114(03)00094-0

The literature acknowledges active research in the 1970s and 1980s, see e.g. the reviews in Blostein and Baird [5] and Carter et al. [7], until the first commercial products in the early 1990s. The success of these works relies on the strong available knowledge (as opposed to other document analysis problems): reasonable number of symbols, strict location of the staff lines, strong rules of music writing. But still, the problem remains difficult and solutions are generally computationally expensive, even in cases of typeset music.

Despite the advances in the field and the available softwares, there are still some unsolved problems, and recognition is not error or ambiguity free. As already mentioned in Blostein and Baird [5] and Ng et al. [17], major problems result from the difficulty to obtain an accurate segmentation into individual meaningful entities. This is due to the printing and digitalization as well as to the high interconnections between musical symbols, and to the variability, from one score to the other, but also within the same score (for instance grouped notes may vary in size and shapes).

A lot of work was dedicated to individual symbol recognition. But such methods are highly prone to errors due to the segmentation steps, for the reasons mentioned above. Therefore the needs for structural information become now well recognized (e.g. [17]). Two different levels can be considered. At the symbol level, structural information deals with the description of a note as a spatial arrangement of different components (head, stem, tail, augmentation dot, etc.) [23]. A second level concerns the relationships between symbols and involves musical rules. Much less approaches have been dedicated to the modeling of structural information at this level and to its use for the recognition. For instance the recognition method developed by Coüasnon and Camillerap [8] is entirely controlled by a grammar which formalizes relative positions between objects. Several works also use metric information and check the note alignment consistency for detecting note length errors and to correct them in some cases [9,6]. Local corrections are also made possible by returning to the low-level processing steps [16,13]. Based on the limits of these local correction possibilities, Kopec et al. [14], Stuckelberg and Doermann [20] express the problem as the global optimization of a functional expressing the likelihood of the interpretation.

Due to the limited work at structural level and to the lack of methods dealing with non-local rules and corrections, we concentrate in this paper on the second level of structural information, with the aim of modeling and using musical rules for disambiguating some recognition hypotheses as well as correcting errors.

Another objective of this paper is to solve ambiguities by accounting for sources of imprecision and uncertainty. Such imperfection may arise at different levels: scanning and segmentation, but can also be intrinsic to the music. For instance the position of a symbol is defined up to some tolerance, some musical rules are not strict, etc. Even strict rules may have to be considered to some degree (for instance the position of accidentals, as explained in Section 2).

Most approaches for dealing with uncertainty and combining it with higher level information are based on statistical methods [1,21] or based on graphs and graph-rewriting rules expressing both binary interactions between symbols and higher-order notational constraints [12]. Here we rely on the fuzzy set and possibility framework, since it offers concepts, tools and properties which are well adapted to the integration of flexible rules and constraints [10] and for dealing with spatial imprecision [15,3,4].

Until now very few work using fuzzy set theory in the optical music recognition domain can be found in the literature. In Watkins [23], the proposed approach uses a fuzzy graph grammar describing the structure of a note, but no structural information at the second level (relationships between symbols). Fuzziness is used for representing concepts like close to (for instance between a note head and stem). In Su et al. [22], a neuro-fuzzy approach is developed for the classification of symbols. No global musical rules are used.

The original contribution of this paper relies in the modeling of structural information and rules using the fuzzy set and possibility theory, and the fusion of such pieces of knowledge in order to improve the decision making step. We make use of various semantics of fuzzy sets or possibility distributions [10]. For instance confidence degree semantics are used in the fusion step, which provides an evaluation of an hypothesis, expressed as an assignment of a group of symbols to recognition classes. Similarity semantics allow us to model symbol classes, by comparison of a symbol to a prototype of each class. Plausibility semantics are used for modeling the relative position of symbols. And preference semantics allow us to model in a simple and efficient way non-mandatory constraints, such as the repetition of an accidental. This is another powerful feature of fuzzy set theory, to be able to model very heterogeneous knowledge in a common mathematical framework, which makes the fusion and decision steps possible and easier [11]. Moreover, the variety of available fuzzy fusion operators is another interesting feature since different pieces of knowledge do not have to play the same role in the fusion [2].

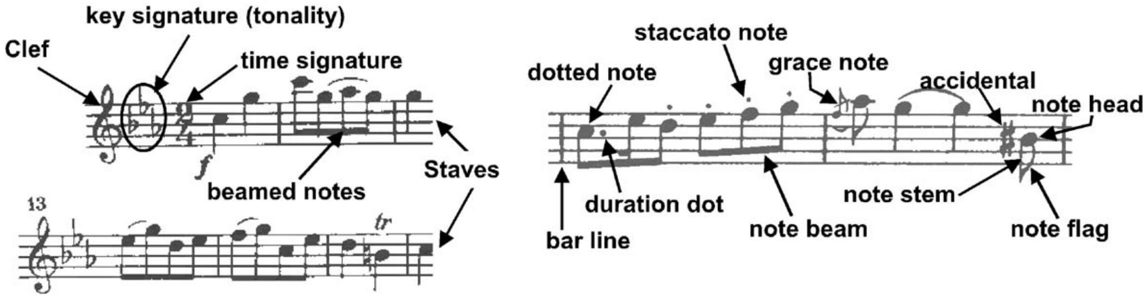
This paper is organized as follows. In Section 2, we recall some basics about music notation and the terminology used in this paper. In Section 3, we briefly summarize the proposed recognition method. Section 4 concerns the first step, the individual symbol analysis. Since it is not the focus of this paper, we refer to [18] for further details on this step. The following sections constitute the core of this paper and the original contribution. Symbol classes are modeled in Section 5. Then several rules are introduced for checking the consistency of hypotheses: graphical consistency is addressed in Section 6 (compatibility between accidental and note, between grace note and note, between note and augmentation dot, etc.), while syntactic consistency is addressed in Section 7 (accidentals, tonality, metric). The proposed decision rule is described in Section 8. Section 9 presents some experimental results, and show that our approach leads to very good results, and is able to correct errors made by other approaches, such as the one of SmartScore.

2. Musical notation—terminology

Before presenting our work, it may be useful to recall some basics of the music writing, define the musical symbols and the terminology used in this paper.

Fig. 1a shows a part of a music sheet. It is composed of several staves. A staff is an arrangement of five parallel equally spaced lines. Musical symbols (Fig. 2) are put on the staff lines. Some global information is indicated at the beginning of every staff: the clef, the tonality indicating which accidentals have to be implicitly applied to the notes of the whole staff, and at the beginning of the first staff, the time signature (metric) indicating the number of beats per bar and the beat value (or pulse). A bar is a set of musical symbols between two bar lines (Fig. 1b). The staff is read from left to right, and the horizontal axis represents the time. The vertical position of a note head related to the staff lines and the key indicates its pitch. Its length is deduced from the number of beams or flags. Accidentals (flat, natural, sharp) are sometimes put before a note head to modify its pitch.

A voice is a musical line, that may correspond to a single instrument. In case of monophonic music, there is only one voice per staff, without any chord (group of notes played together). This



(a) Part of a monophonic music sheet

(b) Two bars

Fig. 1. Music terminology.

𝅝	𝅗𝅥	♩	♪	𝅘𝅥𝅯	𝄻	𝄼	𝄽 𝄾	𝄾	𝄿	•	♯ ♭ ♮	♪	|
1	1/2	1/4	1/8	1/16	1	1/2	1/4	1/8	1/16				
Whole note	Half note	Filled notes: quarter, eighth, sixteenth ... notes			Whole, half, quarter, ...rests					Dot	Accidentals (sharp, flat, natural)	Grace note	Bar line

Fig. 2. Musical symbols and length.

hypothesis excludes also any music written on several staves. As a consequence, there is only one musical symbol at each horizontal position (no superposition of notes for example).

Musical notation is codified by some graphical rules and some syntactic rules. We indicate the most important of them below.

Graphical rules are about the relative positions of the musical symbols:

1. An accidental is placed before a note head and at the same vertical position. Although this expression of this rule is strict, the same position in the image should be understood up to some imprecision, less than the half spacing between two staff lines.
2. A duration dot is placed after a note head.
3. A staccato dot is placed above a note head.

Syntactic consistency is expressed by rules relating symbols to global information on the music sheet, such as the metric and the tonality:

4. The number of beats in a bar must match the time signature (bar length rule).
5. Beams generally bind together notes into discrete groups, generally a whole number of beats or half-beats, so that the beat structure is better isolated.
6. The accidentals of the key signature are applied to every note of the same height up to octave shifts (e.g. every F).
7. An accidental is applied to the following note, but also implicitly to any other notes of the same height (up to octave shifts) present in the remainder of the bar.

8. An accidental may be repeated even if unnecessary in order to make the reading of the score easier.
9. A duration dot multiplies the length of the dotted note by 1.5.

In this paper, we use indifferently the term of “symbol” or “object”, for an entity found during the segmentation process, and to which the recognition process assigns one of the following labels: whole note, half-note, filled note, whole rest, half-rest, quarter rest, eighth rest, sixteenth rest, sharp, flat, natural, grace note, duration dot, bar line (see Fig. 2). We never refer to a subpart of an entity: for example, the stem of a note is not an object, nor a flag.

We say that an object is next another object, when it follows directly this object, in the sense of the natural left to right reading of a monophonic staff. The ordered sequence of symbols is found during the segmentation process, by ordering them by increasing x-coordinate: given an object s_n, the next object is numbered s_{n+1}.

3. General overview

The proposed recognition method aims at automatic reading of printed scores. The score sheets are scanned at the resolution of 300 dpi. This quality provides enough precision for recognition purposes, although some non-trivial problems have to be solved. Having a higher resolution scanning would lead to a heavy memory and computing load, making the whole process less user-friendly without solving some important problems treated in this paper: the defaults of print due to the edition itself, and the variability in the fonts.

Then, the images are binarized to provide an image I where $I(x, y)$ at point (x, y) can take values 0 (white pixels in the following figures) or 1 (black pixels). The binarization process is outside the scope of this paper. Some global information such as the clef, the key-signature, the time signature is also assumed to be known and given as input to the method. The system handles at this time only typeset monophonic music and recognizes the symbols which are essential for reproducing the melody: bar lines, notes with pitch and length value, rests, accidentals, duration dots. It ignores text and annotations below or above the staff.

The processing flow illustrated in Fig. 3 can be divided into three main parts. A symbol analysis process performs the segmentation of the objects (i.e. individual symbols) and provides for each one some recognition hypotheses. An hypothesis is an assignment of a symbol to a class chosen in a set of symbol models. This first step is described in detail in Rossant [18] and summarized in Section 4. The second part of the algorithm consists of a fuzzy modeling step which provides for each classification hypothesis resulting from the first step a possibility degree of membership to the class. It also introduces a fuzzy representation of the common music writing rules by expressing graphical and syntactic compatibility degrees between the symbols. This part uses the global information given as input of the program. This is not a tedious work for the user, and this is typically the type of interaction he is ready to provide, in order to achieve better reliability and better robustness. To relax this assumption and allow changes in time signature or key signature, we can add an additional processing in the first step to recognize them. This has not yet be done. Finally, the decision process evaluates bar per bar all the hypotheses combinations and choose the most consistent one. The fuzzy representation is presented in Sections 5–7, the decision process in Section 8.

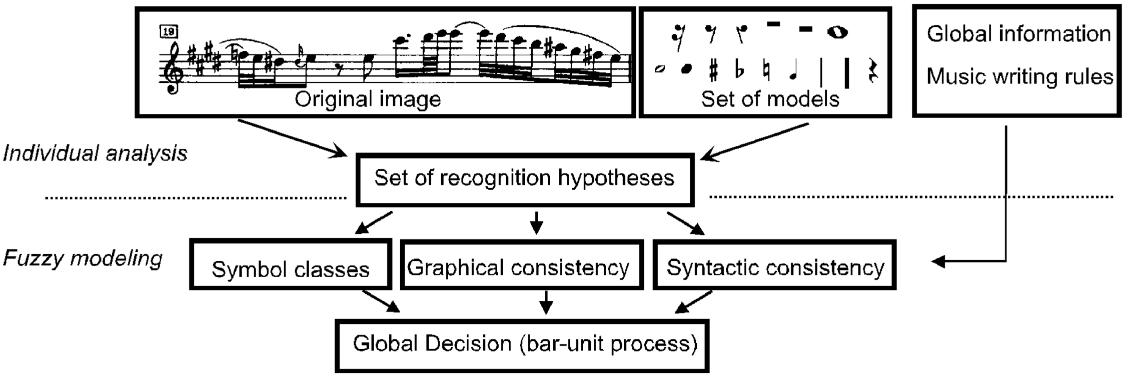


Fig. 3. The global processing flow.

4. Individual symbol analysis

4.1. Method

The first step of the processing consists in segmenting the image into individual symbols s and analyzing each of them separately [18]. This analysis process is mainly based on template matching. We construct a reference base containing a set of models for all symbols. The models are designed for a typical score size (about A4) and for the chosen scanning resolution. This avoids a preliminary scaling step. This is done off line but can be learned or updated for each score based on first recognition results [18]. We compute in a small search area around s the correlation scores between s and each model M^k of the reference base, defined as

$$C_s^k(x,y) = \frac{1}{d_x^k \cdot d_y^k} \sum_{i=0}^{d_x^k-1} \sum_{j=0}^{d_y^k-1} M^k(i,j)\,.\,I'(i,j) \quad (1)$$

with

$$M^k(i,j) = \begin{cases} -1 & \text{for a white pixel,} \\ 1 & \text{for a black pixel,} \end{cases} \quad 0 \leqslant i < d_x^k,\ 0 \leqslant j < d_y^k$$

I', the sub-image of size $d_x^k.d_y^k$ extracted from I around (x,y):

$$I'(i,j) = \begin{cases} -1 & \text{if } I(x+i-i_k, y+j-j_k) = 0, \\ 1 & \text{if } I(x+i-i_k, y+j-j_k) = 1, \end{cases} \quad 0 \leqslant i < d_x^k,\ 0 \leqslant j < d_y^k,$$

where (i_k, j_k) are the coordinates of the center of the model image $M^k(i_k \cong d_x^k/2,\ j_k \cong d_y^k/2)$.

For each model M^k, we compute the highest value $C^k(s) = C_s^k(x_k, y_k) = \max_{(x,y)} C_s^k(x,y)$ obtained at the (x_k, y_k) coordinates. These coordinates represent also the localization of the center of the musical symbol in the image I for the hypothesis of the k class. The models are ranked according to the scores $C^k(s)$. Then a set of rules is used to select for each pattern s at most three recognition hypotheses (H1, H2, H3), each of them assigning the pattern to a possible class. Let us denote

Table 1
Rules for storing recognition hypotheses for object s

	If $C^{k1}(s) \geqslant t_d(k_1)$	If $t_d(k_1) > C^{k1}(s) \geqslant t_m$	If $C^{k1}(s) < t_m$
H1	Class of the model M^{k1}	No symbol (—)	No symbol (—)
H2	Class of M^{k2} if $\begin{cases} t_m \leqslant C^{k2}(s) \\ (C^{k1}(s) - C^{k2}(s)) < t_a \end{cases}$	Class of M^{k1}	No symbol (—)
H3	Class of M^{k3} if $\begin{cases} t_m \leqslant C^{k3}(s) \\ (C^{k1}(s) - C^{k3}(s)) < t_a \end{cases}$	Class of M^{k2} if $\begin{cases} t_m \leqslant C^{k2}(s) \\ (C^{k1}(s) - C^{k2}(s)) < t_a \end{cases}$	No symbol (—)

by $C^{k1}(s)$, $C^{k2}(s)$ and $C^{k3}(s)$ the three highest scores obtained by the models M^{k1}, M^{k2}, M^{k3}, in decreasing order. Based on these scores, the rules are defined in Table 1.

The parameter t_m is the *minimum threshold* which has always to be reached to consider a class as possible, and t_a is the *ambiguity threshold* defined to deal with a secondary highest score close to the first one. We use respectively $t_m = 0.3$ and $t_a = 0.3$. These values have been experimentally optimized. For a bigger t_m value, the right class is more often discarded, and for a smaller t_m value, much more hypotheses are retained, increasing the computation cost of the next processing. The definition of t_a results from a similar compromise. The chosen t_m and t_a values proved to be efficient for all examples we had: right hypothesis exceptionally discarded, number of hypotheses minimized.

The *decision* threshold values $t_d(k)$ are defined for each class k by

$$t_d(k) = \alpha_k * t_d, \quad t_d = 0.5. \tag{2}$$

Experiments show indeed that some musical symbols, for example a sharp, are very sensitive to typewriting variations while some others are much more robust. The probabilities of false detection are not identical for all classes either. That is why the α_k factors have been defined and experimentally optimized, using a number of different scores. For example, the flat symbol does not vary significantly from one publishing to another and gets frequently a high correlation score with patterns not belonging to this class or any other class. That is why we need a high α_k value (1.4) in order to select the flat as first hypothesis (H1) only if the tested object matches closely the model of the reference base and in order to allow the possibility of no symbol in the other cases. On the contrary, α_k is equal to 0.9 for a sharp because the model of the reference base may be quite unsuitable to the analyzed music sheet. All the α_k factors are ranging from 0.8 to 1.5 which guarantees that $t_d(k) \geqslant t_m$ for the chosen t_m value.

For the hypotheses which classify a symbol as a note, the length is obtained by counting the number of flags or beams grouping notes, and from the correlation computed with a possible augmentation dot that may only appear after a note head (see Fig. 4).

4.2. Example

Fig. 5 shows a whole bar extracted from a musical sheet, and the recognition hypotheses are superimposed on the original image in Fig. 6.

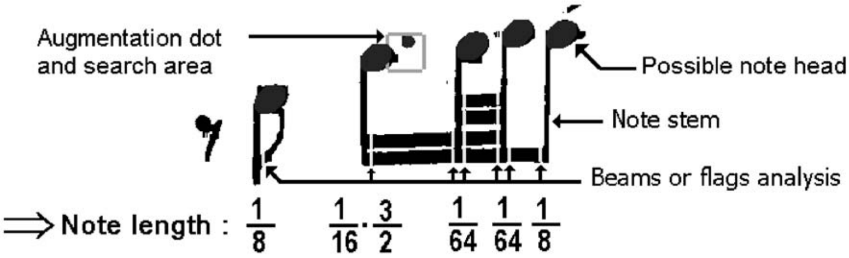

Augmentation dot and search area
Possible note head
Note stem
Beams or flags analysis
⇒ Note length : 1/8 1/16 . 3/2 1/64 1/64 1/8

Fig. 4. Analysis of note length.

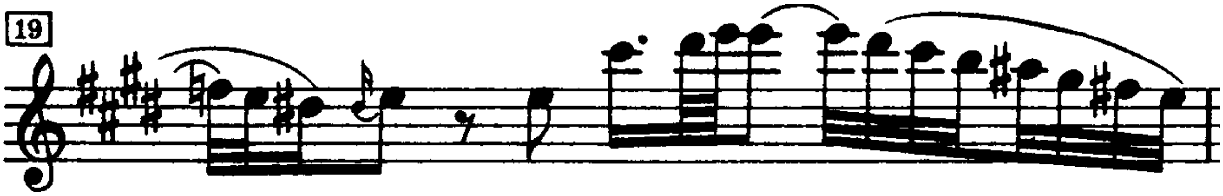


Fig. 5. Original image.

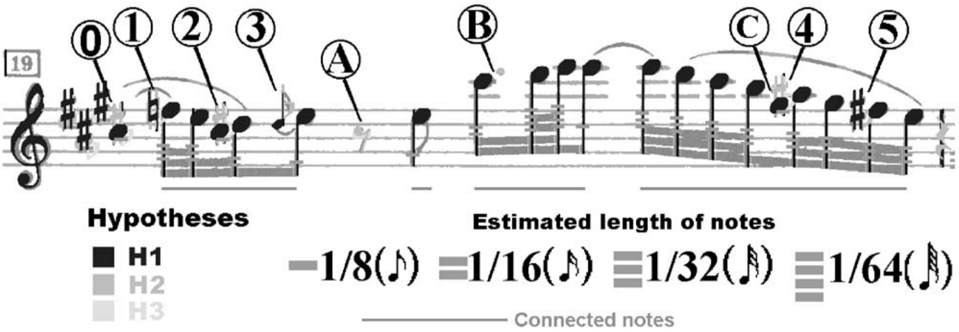


Fig. 6. Results of the individual symbol analysis.

Some of the corresponding recognition hypotheses (— for absence of symbol) and the associated correlation scores are summarized in Table 2. For example, the highest score is obtained for symbol A by the eighth rest but this score is lower than the corresponding decision threshold $t_d(k)$: $t_m < 0.74 < t_d(k)$ as $t_d(k) = 0.75$ for this symbol. So it is according to Table 1 a H2 hypothesis, the H1 hypothesis being 'no symbol' (—), to allow the possibility that there is no symbol at this place.

4.3. Discussion

This example illustrates the limits of the individual analysis. Indeed the decision rule consisting in choosing the class of the model reaching the highest correlation score is not suitable because of the ambiguities between the correlation scores obtained for each object. It is obvious in this example that this method does not lead to the correct solution. But we can assume that we could arbitrate

Table 2
Correlation score resulting from the individual symbol analysis

	0	1	2	3	A	B	C	4	5
H1	● 0.65	♮ 0.71	● 0.62	♩ 0.76	—	—	—	● 0.65	♯ 0.65
H2	♯ 0.62	♯ 0.57	♯ 0.59	♭ 0.49	𝄾 0.74	• 0.93	• 0.55	♯ 0.65	♭ 0.52
H3	♭ 0.57	♭ 0.46	♭ 0.50	-	▬ 0.63			♭ 0.48	—

between the ambiguous hypotheses if we introduce some music writing rules defining relationships between the different symbols present in a bar. The main rules, which are more or less strict, have been expressed in Section 2.

It is obvious that these rules can help in our example. For instance, rules 4, 5 and 9 (see Section 2) favor the recognition of symbol B as a duration dot. According to rule 1, symbol 5 is preferentially recognized as a sharp rather than as a flat because the vertical alignment of the center of symbol 5 and the center of the following note is better for the hypothesis of a sharp than for the hypothesis of a flat. It is also interesting to verify through rules 6, 7, 8 which combinations of accidentals for symbols 1 and 5 are consistent with the tonality.

The following sections aim at modeling the knowledge expressed by these rules and introduce it in the recognition process. It should be noted that the rules have different degrees of flexibility. For instance, rule 4 is strict while rule 8 is loose. Fuzzy set and possibility theory offer a good formal framework for modeling this knowledge and dealing with it.

5. Symbol classes

The first step of the analysis provides similarities between each analyzed symbol and the prototypes (models) of symbol classes, defined as correlation scores. Let $C^k(s)$ the correlation score between object s and the model of class k. The highest the score, the highest the possibility that this object belongs to class k. Therefore we define the degree of possibility $\pi_k(s)$ that s belongs to class k as an increasing function of $C^k(s)$:

$$\pi_k(s) = f_k(C^k(s)), \tag{3}$$

where the shape of this possibility distribution is given in Fig. 7.

The purpose of this transformation is as follows. The correlation scores are often very ambiguous because of segmentation defects but also due to the variations in typewriting. To bypass this problem we propose to use the results output by the analysis process in order to define for each class a possibility distribution which can be interpreted as a similarity between a symbol of the score and the reference model.

The possibility distribution π_k depends on two parameters, S_k which defines the point with the possibility degree equal to 0.5, and D which represents the width of the uncertainty area. Here the parameter D is kept constant. In our experiments, we use $D = 0.3$. Indeed, the correlation score

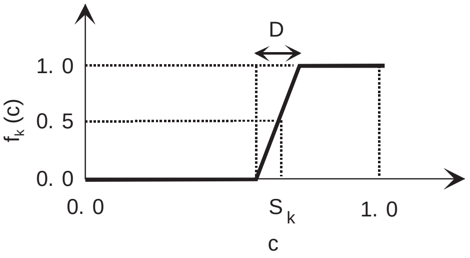


Fig. 7. Possibility distribution of class k as a function of the correlation.

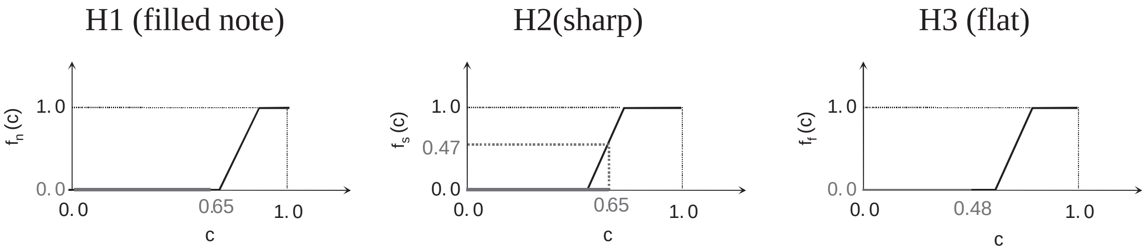


Fig. 8. Possibility degrees computed for object 4.

computed for two different symbols belonging to the same class do typically not differ more than 0.15. By choosing an uncertainty area twice larger, we verify in our experiments that the degrees of possibility get a good dynamic. But this parameter still needs to be further optimized.

The parameter S_k is learned from the results of the first analysis step. Let $n(k)$ be the number of objects having a correlation score larger than the threshold value $t_d(k)$, and let $m(k)$ be the average value of the scores of the objects. We defined S_k as

$$S_k = \frac{t_d(k) + n(k)m(k)}{n(k) + 1} + \frac{D}{2}. \tag{4}$$

The meaning of this parameter is as follows. Let us assume for instance that all $n(k)$ objects have exactly the score $t_d(k)$. Then $S(k) = t_d(k) + D/2$, meaning that the possibility is zero until $t_d(k)$, starts to increase after this value, and reaches the value of 1 for $t_d(k) + D$. If $m(k)$ is larger than $t_d(k)$, then the curve is shifted to the right.

Fig. 8 shows three possibility distributions deduced from the first analysis of the whole sheet and illustrates how they are applied to each recognition hypothesis made on object 4 (see also Fig. 6 and Table 2).

The shape of the function π_k does not need to be estimated very precisely. What is important is that it is not a binary function, and that the rank is preserved (a symbol with a higher correlation score to a class has a higher degree of possibility of belonging to this class). Experimentally we observed a good robustness with respect to this shape. This type of robustness has already been experimented in several other applications of fuzzy set theory [10]. It holds also in the next steps of our approach, when modeling the other pieces of knowledge.

6. Graphical consistency

Until now, each object was processed individually. In this section we introduce graphical relationships between two successive objects. Musical writing rules impose some compatibility of position between a note and its accidental, between a note and a grace note, or between an augmentation dot, the dotted note and the following note. Due to possible imprecision in the score and in the segmentation, these rules cannot be used in a crisp way, and are rather a matter of degree. Therefore we define compatibility degrees to express these consistency rules.

6.1. Compatibility between accidental and note

An accidental should be placed before a note and at the same height. Small variations in its horizontal and vertical positions may arise, depending on the density of symbols in the score and on the precision of the location after the segmentation process. Let us assume that object s_n is an accidental belonging to class k, and that the next object s_{n+1} is a note belonging to class k'. The degree of possibility of this hypothesis is a function of the compatibility degree between both symbols, noted $C_p(s_n^k, s_{n+1}^{k'})$ and computed as follows. Let Δl be the difference in horizontal position between s_n and s_{n+1} and Δh the difference in vertical position (see Fig. 9). The admissible values for these two differences are defined by two functions f_l and f_h illustrated in Fig. 10. They depend on the space (the separation between two staff lines), which is a known parameter after the segmentation step.

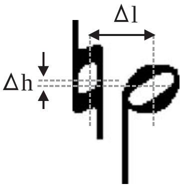


Fig. 9. Horizontal and vertical differences between an accidental and a note.

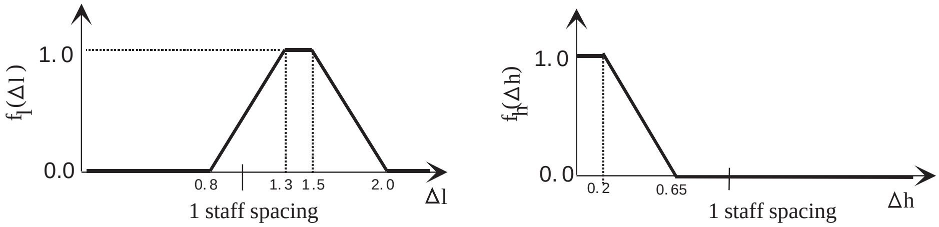


Fig. 10. Admissible values of Δl (left) and Δh (right) in the case of accidentals.

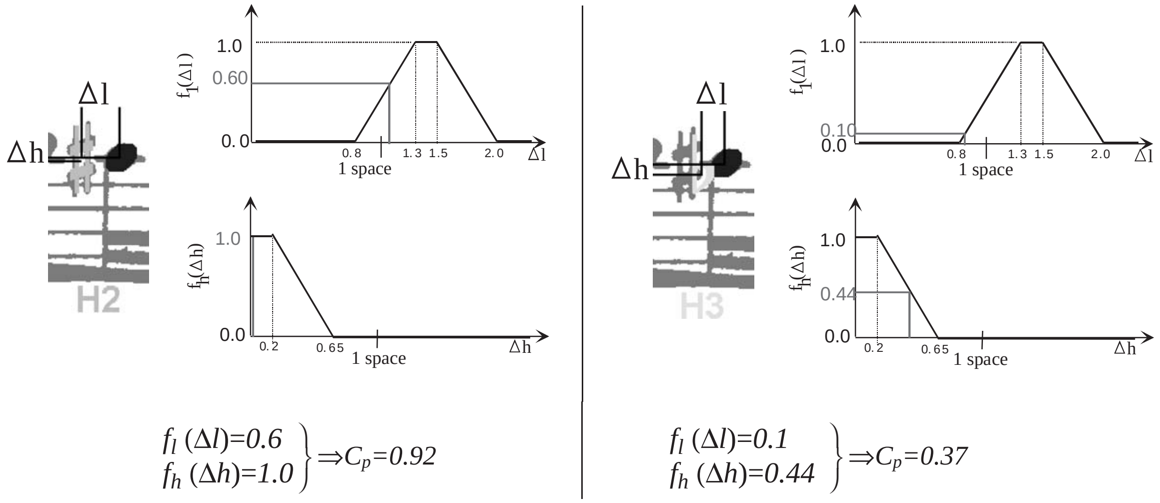


Fig. 11. Graphical compatibility coefficients computed for object 4. Left: evaluation of a sharp hypothesis, right: evaluation of a flat hypothesis, according to the positions found in the individual analysis process.

Then we define:

$$C_p(s_n^k, s_{n+1}^{k'}) = \begin{cases} \alpha_l f_l(\Delta l) + \alpha_h f_h(\Delta h) & \text{if } f_l(\Delta l) > 0 \text{ and } f_h(\Delta h) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

This combination is a compromise between two criteria, excluding the cases where one of the criteria at least is not satisfied at all. Using a degree between 0 and 1 instead as a crisp threshold on each criterion allows us not to discard completely an accidental which is not exactly at the theoretically expected position. This degree behaves monotonically, in the sense that if the relative position of the accidental with respect to the note gets worse, then the degree decreases. The chosen coefficients $\alpha_l = 0.2$ and $\alpha_h = 0.8$ used in the weighted average express the relative importance of the two criteria. Indeed the horizontal shift is not as significant as the vertical one, because a false recognition hypothesis can easily get a perfect horizontal compatibility coefficient. See for example object 3 (Fig. 6) classified as a flat in the second hypothesis instead of as a grace note. So the horizontal compatibility coefficient helps to compare two competitive hypotheses, for example sharp and flat for object 4 (see Fig. 11). We compute for these two hypotheses the horizontal and vertical shifts between object 4 and the following note, according to the positions found in the individual analysis process. We obtain then a compatibility degree of 0.92 for a sharp, and 0.37 for a flat. Giving a greater impact to the vertical compatibility coefficient allows us to rank hypothesis H1 (sharp) before hypothesis H2 (flat) and to reinforce the difference between these two hypotheses.

6.2. Compatibility between grace note and note

The compatibility between a grace note and the next note is defined as the one between an accidental and a note. Only the function f_h is slightly different, as illustrated in Fig. 12. This

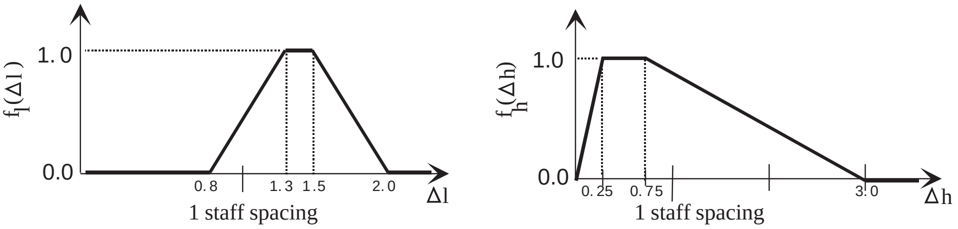


Fig. 12. Admissible values of Δl (left) and Δh (right) in the case of grace notes.

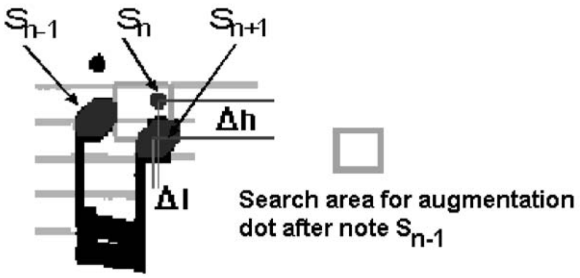


Fig. 13. Search area for an augmentation dot after note s_{n-1} and relative position to the following note s_{n+1}. (When numbering the symbols, a dot found in the search area of a note s_{n-1} is always numbered s_n whatever the position of the next object, here a note.)

function expresses that a grace note is mostly expected at one half-space from the note, but that larger shifts are also possible.

The weight are here chosen as $\alpha_l = 0.5$ and $\alpha_h = 0.5$ representing the equal importance of both criteria.

6.3. Compatibility between note and augmentation dot

During the first analysis process, augmentation dots are searched for in a small area next to the note head (see Fig. 13). Let s_{n-1} be the dotted note, and s_n the augmentation dot found in its search area. We do not express any compatibility degree between these two objects, because all the locations of the dot inside the search area are assumed to be equally admissible. But some confusion between an augmentation dot and a staccato dot is possible if the note s_{n-1} is also followed by a note s_{n+1} and if the dot s_n is above the s_{n+1} note head. Therefore we define a compatibility degree between the hypothesis that s_n is an augmentation dot and the hypothesis that the next object s_{n+1} is a note. The shape of the admissible values for Δl and Δh, the horizontal and vertical shifts between the dot s_n and the note s_{n+1}, is designed in order to avoid any confusion with a staccato dot (Fig. 14).

Then, the compatibility between the hypothesis that s_n is an augmentation note and s_{n+1} a note is defined as

$$C_p(s_n^k, s_{n+1}^{k'}) = \mathrm{Max}[f_l(\Delta l), f_h(\Delta h)] \tag{6}$$

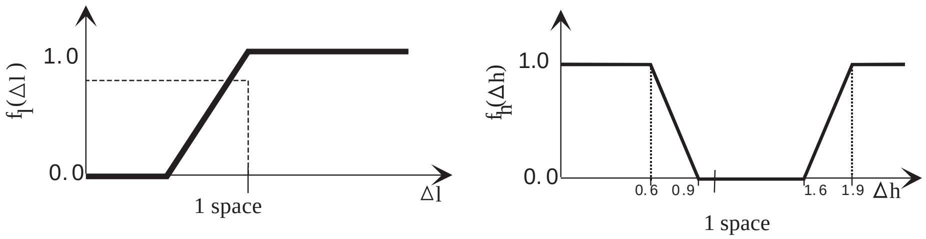


Fig. 14. Admissible values of Δl (left) and Δh (right) in the case of augmentation dots.

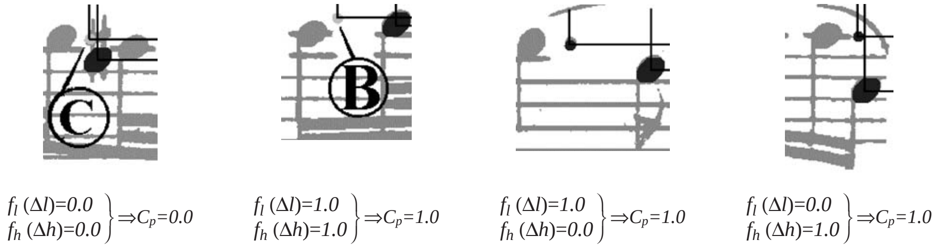

C
B
f_l (Δl)=0.0, f_h (Δh)=0.0 ⇒C_p=0.0
f_l (Δl)=1.0, f_h (Δh)=1.0 ⇒C_p=1.0
f_l (Δl)=1.0, f_h (Δh)=0.0 ⇒C_p=1.0
f_l (Δl)=0.0, f_h (Δh)=1.0 ⇒C_p=1.0

Fig. 15. Some examples of compatibility between a dot and the next note.

expressing that the compatibility should be high as soon as one of both criteria is well satisfied. Fig. 15 illustrates why a more indulgent rule than for accidental and grace note has been chosen.

In the first case both Δl and Δh fall outside the admissible range, leading to a compatible degree equal to 0. This result reflects that the dot would be interpreted as a staccato dot rather than as an augmentation dot. In the second case, both values are admissible, which yields a full compatibility. In the two last cases, only one criterion is well satisfied while the second is not. In both cases the resulting compatibility degree is equal to 1, which fits what was intuitively expected in these cases.

6.4. Compatibility between any two symbols

For all other configurations, such as a note followed by a note, or a rest followed by a note, no specific music rule depending on the class of the symbols can be expressed. However, only one musical symbol is vertically expected per staff in case of monophonic music. So we can again define a graphical compatibility degree between two consecutive symbols s_n and s_{n+1} as a function of Δl, the difference in horizontal position between the centers of s_n and s_{n+1} (see Fig. 16).

The typical width of a musical symbol being greater than the staff spacing, the function reaches the maximum value for Δl equal to one space. Intermediate Δl values between the half-space and the space allow to account for segmentation imprecision. Then, the degree of possibility that the object s_n belonging to class k is followed by the object s_{n+1} belonging to class k' is expressed by the following graphical compatibility coefficient:

$$C_p(s_n^k, s_{n+1}^{k'}) = f_l(\Delta l). \tag{7}$$

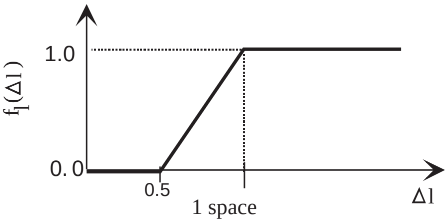
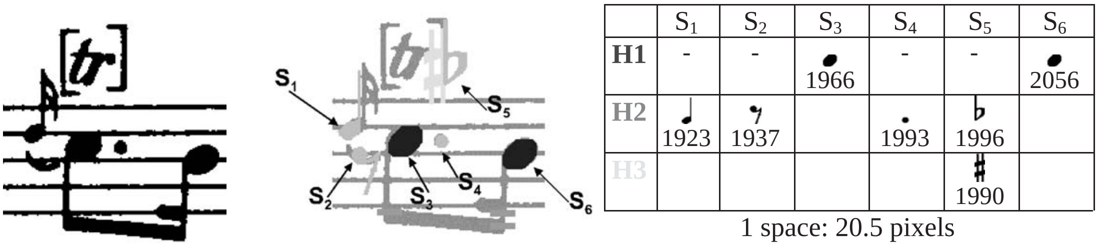


Fig. 16. Admissible values of Δl for two any symbols.



	S1	S2	S3	S4	S5	S6
H1	-	-	● 1966	-	-	● 2056
H2	♩ 1923	𝄾 1937		• 1993	♭ 1996	
H3					♯ 1990	

1 space: 20.5 pixels

Fig. 17. An example of hypotheses with their horizontal coordinates.

Fig. 17 shows a part of a music score, and the recognition hypotheses made on the objects with their horizontal coordinates in the score sheet.

One possible hypothesis grouping consists in classifying S_1 as a grace note, S_2 as an eighth rest, S_3 and S_6 as notes, S_4 as a dot, S_5 as a flat. The horizontal difference between S_1 and S_2 is equal to 0.68 staff spacing and leads to a graphical compatibility coefficient equal to 0.36; in the same way, we compute a compatibility coefficient between S_2 and S_3 equal to 1.0, and a compatibility coefficient between $S_4$and S_5 equal to 0.0. Two of these values are lower than 0.5, reflecting two conflicts in the horizontal location of the objects. A second configuration suppressing the objects S_2 and S_5 solves this problem. This time, according to Section 6.2 and Eq. (5), we compute a compatibility degree between the grace note S_1 and the note S_3 and according to Section 6.3 and Eq. (6) a compatibility degree between the dot S_4 and the following note S_6. Both values are equal to 1.0 reflecting the perfect graphical compatibility of this configuration. A third configuration suppressing S_4 and keeping S_5 as an accidental would also solve one of the conflicts. But this time we would have to compute a compatibility degree between an accidental (S_5) and a following note (S_6), which would be equal to 0.0 according to Section 6.1 (Eq. (5)). So we can see on this example how the graphical compatibility of a set of recognition hypotheses can be evaluated, favoring consistent configurations.

7. Syntactic consistency

In this section, we introduce the syntactic musical rules relating symbols to global information on the music piece. Some of these rules are strict, other are more flexible and aim at making the

Table 3
Compatibility coefficients between two accidentals in a bar

	$S_n=$ sharp	$s_n=$ natural	$s_n=$ flat
$S_m=$ void	0.75	0.5	0.75
$S_m=$ sharp	0.5	1.0	0.0
$S_m=$ natural	1.0	0.5	1.0
$S_m=$ flat	0.0	1.0	0.5

music reading easier. Three types of rules are introduced here, related to the accidentals indicating the tonality, related to links between the tonality and accidentals, and related to the metric.

Each set of recognition hypotheses is evaluated against each rule, and a possibility degree is assigned to each symbol of the set concerned by this rule. This degree represents a compatibility coefficient between the symbol, the other symbols of the set, and the rule.

7.1. Tonality accidental

The tonality is given as an input in our recognition procedure. A strict rule expresses that the accidentals at the beginning of the staff should correspond to this tonality, as a fixed sequence of sharps or flats. To any tonality accidental s_n of class k satisfying this rule a compatibility coefficient $C_s(s_n^k)$ equal to 1 is assigned, while a coefficient equal to 0 is assigned to any others not satisfying this rule. These coefficients are binary possibility degrees. This means that the later hypotheses will be completely discarded in the following. In Fig. 6, the object number 0 gets a coefficient equal to 1 for the hypothesis of a sharp, and a coefficient equal to 0 for the hypothesis of a flat. Hypothesis H1 (note head) is not evaluated according to this rule since it does not involve an accidental.

7.2. Links between tonality and accidentals

If a symbol s_n possibly belongs to one of the accidental classes, according to the first analysis step, then it should be consistent with the key signature and with other accidentals in the same bar or in the bars before (see Section 2, rules 6,7,8).

Let us consider the case where there is no accidental at the same height as s_n in the key, and let s_m be a previous accidental at the same height as s_n (up to octave shifts) and in the same bar. This configuration may have different compatibility degrees, depending on the type of s_n and s_m (sharp, flat, or natural). Table 3 summarizes the chosen degrees $C_s(s_n^k)$ attributed to s_n:

These degrees have been defined as follows: the most common configurations are when a sharp or a flat occurs for the first time in the bar, or when a natural cancels a previous flat or sharp. The first configuration gets a possibility degree equal to 0.75, so over the middle value while the second configuration gets a degree equal to 1.0 in order to strengthen any consistent interaction bringing new information inside a bar. But it is also possible that the second accidental recalls the first one in order to make the reading easier, although it is theoretically not necessary. That is why the corresponding degree takes the middle value 0.5 reflecting that this configuration is possible but

Table 4
Syntactic compatibility between two accidentals inside a bar with a sharp in the key signature

	$S_n =$ sharp	$s_n =$ natural	$s_n =$ flat
$S_m =$ void	0.5	1.0	0.0
$S_m =$ sharp	0.5	1.0	0.0
$S_m =$ natural	1.0	0.5	0.0
$S_m =$ flat	0.0	1.0	0.5

Table 5
Compatibility coefficient between two accidentals in two different bars

	$S_n =$ sharp	$s_n =$ natural	$s_n =$ flat
$S_m =$ sharp	0.5	0.5	0.0
$S_m =$ natural	0.5	0.5	0.5
$S_m =$ flat	0.0	0.5	0.5

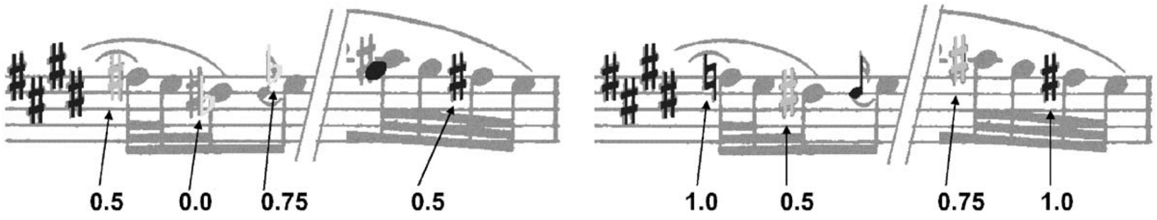


Fig. 18. Two examples of compatibility coefficients between tonality and accidentals: on the left, the accidental hypotheses are less consistent with the tonality than on the right.

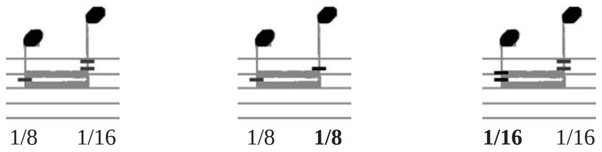
corresponds to an usual and non-obligatory practice. The last possibility degree is 0 when a flat occurs after a sharp, or the contrary, because this configuration should not occur.

Table 4 provides the new possibility degrees in the case where there is a sharp in the key signature at the same height as s_n and s_m.

The possibility degree has been changed following a similar reasoning: 0.0 for an exceptional configuration (e.g. a flat when a sharp in the key signature), 0.5 for an usual practice but not mandatory (e.g. the recall of the sharp already in the key signature), 1.0 for a consistent association bringing new information (e.g. a natural when a sharp in the key signature).

Lastly, Table 5 indicates the compatibility coefficients when s_m is a few bars before s_n, making this time an association such as sharp/natural or sharp/sharp equally possible and an association such as flat/sharp impossible.

Fig. 18 illustrates the mechanism of these coefficients on two different hypothesis groupings, assuming that there is no accidental at the same height in the previous near bars. It shows how distant objects interact together.



(a) Hypothesis H^{l0} (initial) length = 3/16 ($C_l^{H^{l0}} = 1.0$)

(b) Hypothesis H^{l1} length=1/4 ($C_l^{H^{l1}} = 0.5$)

(c) Hypothesis H^{l2} length=1/8 ($C_l^{H^{l2}} = 0.5$)

Fig. 19. Example of length hypotheses with their possibility degrees (without any augmentation dot in the set of the recognition hypotheses).

7.3. Metric

Finally metric rules are introduced. One strict rule is the number of beats per bar. An hypothesis consisting of an assignment of all symbols found in a bar is valid (will have a possibility degree equal to 1) if it satisfies this rule.

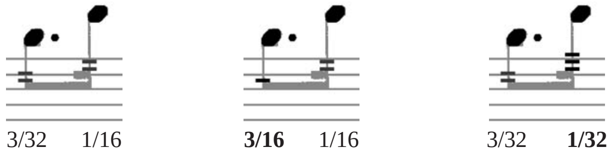
A less strict rule concerns groups of filled notes. For instance eighth notes are usually grouped in order to build one beat. The individual first analysis process provided a length hypothesis for each note, which is now revised considering the common grouping conventions. The beamed notes are extracted through a region growing algorithm and their rhythmical internal organization compared with the usual ones, according to the time signature. Then, at most two hypotheses are made for each note, increasing or decreasing the total length of the group, and changing the smallest number of values. Let $L(g)$ be the number of notes in the group g, and $l(g)$ the number of length changes. The possibility degree affected to the whole group for the hypothesis H^l is computed by

$$C_l^{H^l}(g) = 1.0 - \frac{l(g)}{L(g)}. \tag{8}$$

The meaning of this possibility degree is different from the previous ones. Indeed, it does not evaluate directly an hypothesis against a musical rule, but against the initial interpretation of the note lengths, which is assumed to be reliable. So, we use a musical rule, which is not a strict one, by suggesting some possible corrections, but the more the new interpretation differs from the initial one, the more the possibility degree decreases.

The set of the admissible length values is conditioned by the augmentation dots that are in the set of recognition hypotheses. That means that the length of a non-dotted note can only take its length value in the set $\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \dots\}$ while a dotted note can only take its length value in the set $\{\frac{3}{8}, \frac{3}{16}, \frac{3}{32}, \frac{3}{64}, \dots\}$.

Fig. 19 illustrates this process on a simple example of two beamed notes, assuming that the first one is misinterpreted. The initial configuration represented in (a) gets an unusual total length. The two closest hypotheses that can be made are represented in (b) and (c), with their possibility degree (lower than in (a) according to Eq. (8), since the new interpretation differs from the initial one). Another hypothesis, such as 3/16, 1/16, leads also to an usual length of group. But it is not considered because it would make the assumption that the first note is a dotted one.

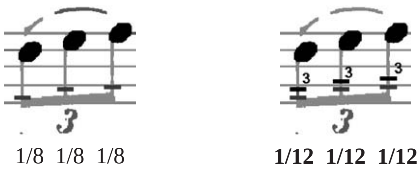


(a) Hypothesis H^{l0} (initial) length = 5/32 ($C_l^{H^{l0}} = 1.0$)

(b) Hypothesis H^{l1} length=1/4 ($C_l^{H^{l1}} = 0.5$)

(c) Hypothesis H^{l2} length=1/8 ($C_l^{H^{l2}} = 0.5$)

Fig. 20. Example of length hypotheses with their possibility degrees (with an augmentation dot in the set of the recognition hypotheses).



(a) Hypothesis H^{l0} (initial) length = 3/8 ($C_l^{H^{l0}} = 1.0$)

(b) Hypothesis H^{l1} (triplet) length=1/4 ($C_l^{H^{l1}} = 1.0$)

Fig. 21. Example of length hypotheses with their possibility degrees in case of a triplet.

Fig. 20 illustrates another simple configuration, with a dot augmenting the first note. This time, a length hypothesis such as 1/16 is not considered because it is not possible with the dot.

This method allows us also to detect the triplets occurring sometimes in binary metric. For example, if the algorithm detects three beamed notes with one beam for each one, the total length of the group may be 3/8 or 1/4 in case of a triplet. That is why this second hypothesis is also suggested, with a possibility degree equal to 1.0, because it does not result from a misunderstanding of the number of beams (Fig. 21).

This method has proven to be efficient, especially for the detection of triplets and for the correction of some note length errors due to local segmentation defects.

8. Global decision

The previous modeling provides a way to compute degrees of possibility for hypotheses expressed as an assignment to classes of a symbol or a set of symbols, according to several criteria and musical rules. The next step consists in merging all these criteria in order to make a decision. This decision is based on the search for the optimal configuration according to all criteria. The global optimization problem is divided into sub-problems, where optimization is performed in each bar separately. A configuration j in a bar is composed of a set of $N(j)$ objects s_n $(n = 1..N(j))$ assigned to classes $k(n,j)$. For this configuration, several length hypotheses H^l are also made, combining together the different hypotheses made on each of the $N(j,H^l)$ groups g $(g = 1..N(j,H^l))$ of notes. Such a configuration will be referenced as (j,H^l) in what follows. The decision process is divided into three steps:

- Check for general consistency of the configuration.
- Combination of all the compatibility coefficients and possibility degrees.
- Maximization of the resulting global function expressing the consistency between the symbols and the consistency of the length hypotheses.

These three main steps are detailed below.

8.1. Consistency check

We say that the configuration j is consistent if it verifies the following rules:

- An augmentation dot is in the search area of a note included in the configuration.
- Tonality accidental excepted, the object following an accidental must be a note.
- No compatibility degree is zero.

These rules are used in a crisp way so that inconsistent configurations are immediately discarded.

8.2. Fusion of possibility degrees and compatibility coefficients

For an object n classified in the configuration j as an accidental of class $k(n,j)$, the global compatibility coefficient $C_t^{(j)}$ $(s_n^{k(n,j)})$ is deduced from the average of the graphical compatibility coefficient and the syntactic compatibility coefficient respectively defined in Sections 6.1 and 7.2:

$$C_t^{(j)}(s_n^{k(n,j)}) = \tfrac{1}{2}[C_p(s_n^{k(n,j)}, s_{n+1}^{k(n+1,j)}) + C_s(s_n^{k(n,j)})]. \tag{9}$$

Because grace notes are generally easily confused with accidentals, the global compatibility coefficient is for this class computed in the same way, with a syntactic compatibility coefficient always equal to 0.5:

$$C_t^{(j)}(s_n^{k(n,j)}) = \tfrac{1}{2}[C_p(s_n^{k(n,j)}, s_{n+1}^{k(n+1,j)}) + 0.5]. \tag{10}$$

For tonality, the global compatibility coefficient is equal to the binary syntactic compatibility coefficient C_s $(s_n^{k(n,j)})$ defined in Section 7.1. For augmentation dots followed by a note, it is equal to the graphical compatibility coefficient $C_p(s_n^{k(n,j)} s_{n+1}^{k(n+1,j)})$ defined in Section 6.3.

In any other case, the global compatibility coefficient is equal to the horizontal compatibility degree $C_p(s_n^{k(n,j)}, s_{n+1}^{k(n+1,j)})$ defined in Section 6.4 (Eq. (7)).

Then the global function $Conf_r(j)$ merges the possibility degrees $\pi_{k(n,j)}(s_n^{k(n,j)})$ (Section 5, Eq. (3)) and the compatibility coefficients $C_t^{(j)}(s_n^{k(n,j)})$ of all the recognition hypotheses belonging to the configuration j:

$$Conf_r(j) = \frac{1}{N(j)} \sum_{n=1}^{N(j)} \left[\frac{\pi_{k(n,j)}(s_n^{k(n,j)}) + C_t^{(j)}(s_n^{k(n,j)})}{2} \right]. \tag{11}$$

It expresses the possibility degree for the recognition hypothesis grouping j. Then we have to combine together the hypotheses made on the length of the beamed notes, in order to express the global possibility degree of the note groupings. The global coefficient $Conf_l\ (j,H^l)$ is computed for each configuration (j,H^l) as

$$Conf_l(j,H^l) = \left[\frac{1}{N(j,H^l)}\sum_{g=1}^{N(j,H^l)} C_l^{(j,H^l)}(g)\right]\left[1-\frac{N(j,H^l)}{N'(j,H^l)}\right], \tag{12}$$

where $N(j,H^l)$ is the number of groups of beamed notes and $N'(j,H^l)$ the number of beamed notes. The first term expresses the average of the C_l coefficients computed on each group of notes according to Section 7.3 (Eq. (8)). It is multiplied by a second factor which takes high values when the beamed notes are grouped into few groups. This factor helps to exclude configurations where the false recognition of a note breaks a consistent group of beamed notes in two.

The final function combining all the possibility degrees and compatibility coefficients for the configuration (j,H^l) is given by

$$Conf(j,H^l) = Conf_r(j) * Conf_l(j,H^l). \tag{13}$$

It is the product of two factors, expressing that both criteria, the consistency of the recognition hypotheses and the possibility degree of the length hypotheses, have to be simultaneously satisfied. The use of the product t-norm instead of the minimum for instance makes this rule more severe.

The total length of the bar, noted $D(j,H^l)$, is the sum of the length of the groups of notes in the configuration (j,H^l), and of the length of the rests (only depending on j). The strict rule concerning the number of beats in the bar is expressed below in the third step of the decision algorithm.

8.3. Decision making

The decision algorithm chooses the configuration (j,H^l) which meets at best two decision criteria, which are by priority order:

- The total length $D(j,H^l)$ of the bar is correct.
- The $Conf(j,H^l)$ function is maximized.

This means that the algorithm chooses among the configurations matching the time signature the one which reaches the highest score $Conf(j,H^l)$. If no configuration achieves the length constraint, the algorithm retains the configuration maximizing $Conf(j,H^l)$.

8.4. Example

We will now illustrate on our example how this decision algorithm works. In order to make the analysis more readable, we will focus on the objects which are indexed in Fig. 6. The key signature given as input to the program is 4/4 (4 beats per bar, beat value equal to 1/4). Table 6 summarizes the possibility degrees of membership to a class, for each recognition hypothesis made on the objects.

The number of possible configurations is the product of the number of hypotheses made on each object. But most of these configurations are immediately discarded, because at least one compatibility coefficient is zero. For example, the configuration j_1 illustrated in Fig. 22 and Table 7 is not evaluated because three of the objects get a compatibility coefficient $C_t^{(j)}(s_n^{k(n,j)})$ equal to 0.

Table 6
Degrees of possibility

	0	1	2	3	A	B	C	4	5
H1	● 0.00	♮ 0.60	● 0.00	♩ 0.50	—	—	—	● 0.00	♯ 0.47
H2	♯ 0.37	♯ 0.20	♯ 0.27	♭ 0.00	𝄾 0.80	. 0.80	. 0.10	♯ 0.47	♭ 0.00
H3	♭ 0.00	♭ 0.00	♭ 0.00	—	▬ 0.10			♭ 0.00	—

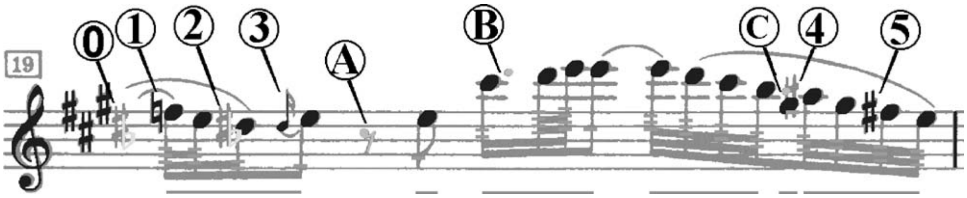


Fig. 22. Configuration j_1.

Table 7
Configuration j_1

$s_n^{k(n,j_1)}$	0 ♭	1 ♮	2 ♭	3 ♩	A 𝄾	B .	C .	4 ♯	5 ♯
$\pi_{k(n,j_1)}\ (s_n^{k(n,j_1)})$	**0.00**	**0.60**	**0.00**	**0.50**	**0.80**	**0.80**	**0.10**	**0.47**	**0.47**
$C_p\ (s_n^{k(n,j_1)},\ s_{n+1}^{k(n+1,j_1)})$		1.0	0.00	0.80	1.0	1.0	0.00	0.92	0.98
$C_s(s_n^{k(n,j_1)})$	0.00	1.0	0.00					0.75	1.0
$C_t^{(j)}(s_n^{k(n,j)})$	**0.00**	**1.0**	**0.00**	**0.65**	**1.0**	**1.0**	**0.00**	**0.83**	**0.99**

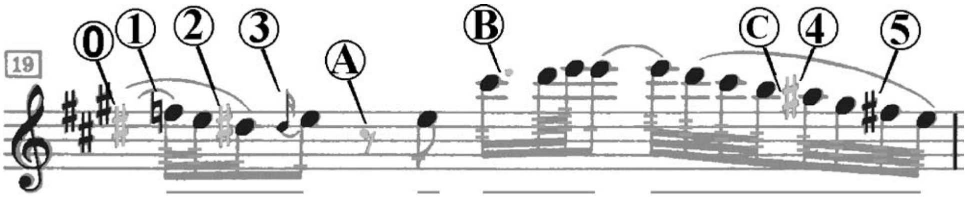


Fig. 23. Configuration j_2 (correct).

The configuration j_2, which is the right solution, is represented in Fig. 23 and Table 8.
The algorithm runs as follows:

- In a first step, it computes the $Conf_r(j_2)$ coefficient, expressing the global degree of possibility of this configuration assigning $N(j_2)=29$ symbols to classes (the last bar line is only used to evaluate the graphical compatibility of the symbol just before it). The program outputs 0.708.

Table 8
Configuration j_2

$s_n^{k(n,j_2)}$	0 ♯	1 ♮	2 ♯	3 ♩	A 𝄾	B •	C -	4 ♯	5 ♯
$\pi_{k(n,j_2)}(s_n^{k(n,j_2)})$	**0.37**	**0.60**	**0.27**	**0.50**	**0.80**	**0.80**		**0.47**	**0.47**
$C_p(s_n^{k(n,j_2)}, s_{n+1}^{k(n+1,j_2)})$		1.00	0.96	0.80	1.0	1.0		0.92	0.98
$C_s(s_n^{k(n,j_2)})$	1.00	1.00	0.50					0.75	1.0
$C_t^{(j_2)}(s_n^{k(n,j_2)})$	**1.00**	**1.00**	**0.73**	**0.65**	**1.00**	**1.00**		**0.83**	**0.99**

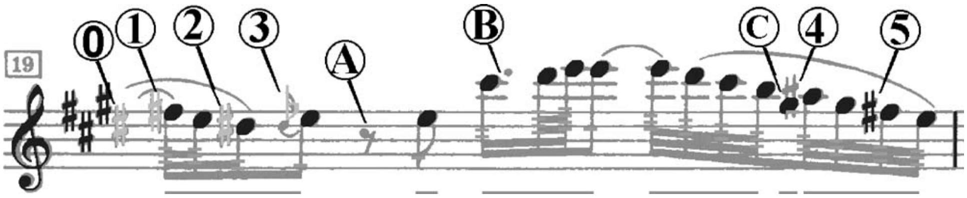


Fig. 24. Configuration j_3.

Table 9
Configuration j_3

$s_n^{k(n,j_3)}$	0 ♯	1 ♯	2 ♯	3 ♭	A -	B -	C-	4 ●	5 ♯
$\pi_{k(n,j_3)}(s_n^{k(n,j_3)})$	**0.37**	**0.20**	**0.27**	**0.00**				**0.00**	**0.47**
$C_p(s_n^{k(n,j_3)}, s_{n+1}^{k(n+1,j_3)})$		1.00	0.96	0.38				1.00	0.98
$C_s(s_n^{k(n,j_3)})$	1.00	0.50	0.50	0.75					0.5
$C_t^{(j_3)}(s_n^{k(n,j_3)})$	**1.00**	**0.75**	**0.73**	**0.56**				**1.00**	**0.74**

- Then, the algorithm examines the beamed notes. There are four groups of beamed notes and every one gets an usual length (one beat or half a beat for the isolated note). There is consequently only one length configuration H^{l0} with a score $Conf_l(j_2, H^{l0})$ equal to 0.765 (every $C_l^{(j_2,H^{l0})}(g)$ is equal to 1.0, and there are $N(j_2, H^{l0}) = 4$ groups of notes for $N'(j_2, H^{l0}) = 17$ notes).
- The total length of the bar is correct: $D(j_2, H^{l0}) = 4$ *beats* and the global score is equal to $Conf(j_2, H^{l0}) = 0.708 * 0.765 = 0.542$.

It is interesting to compare this result with another admissible configuration, such as the one illustrated in Fig. 24 and Table 9 .

- The $Conf_r(j)$ value is lower for j_3 than for the previous configuration j_2 and equal to 0.665. Indeed, if we compare Tables 8 and 9, we can see that the $\pi_{k(n,j)}$ possibility degrees are rightly

Table 10
Length hypotheses made for configuration j_3

	g					
	1	2	3	4	5	6
H^{l0}	1.0 (1/4)	1.0 (1/8)	1.0 (7/32)	1.0 (1/8)	1.0 (1/8)	1.0 (1/8)
H^{l1}	1.0 (1/4)	1.0 (1/8)	0.5 (1/4)	1.0 (1/8)	1.0 (1/8)	1.0 (1/8)
H^{l2}	1.0 (1/4)	1.0 (1/8)	0.75 (1/8)	1.0 (1/8)	1.0 (1/8)	1.0 (1/8)

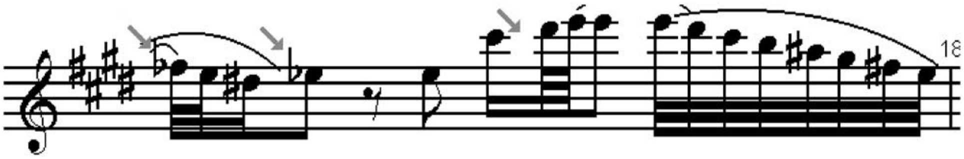


Fig. 25. Result provided by SmartScore (3 errors).

favorable to configuration j_2 and that this model differentiates the two configurations much better than the use of the correlation scores could. Moreover, the $C_t^{(j_3)}(s_n^{k(n,j_3)})$ coefficients are as well acting in this direction. It is also interesting to notice how distant objects interact. For example, the choice of a natural for object 1 strengthens the choice of a sharp for object 5.

- There are this time 6 groups of notes. All of them have an usual length, a binary fraction of the beat, excepted the third one (1/16, 1/64, 1/64, 1/8), because of the absence of the duration dot B. The closest usual values for four beamed notes are: 1/16, 1/32, 1/32, 1/8 or 1/16, 1/64, 1/64, 1/32. The other groups of notes are supposed to be right, and no new hypotheses are made on them. So there are in total three length configurations H^l for j_3. Table 10 summarizes the different coefficients computed for each of them. Each box indicates the $C_l^{(j_3,H^l)}$ (g) coefficient obtained for the group g in the hypothesis H^l, and in brackets, the length of the group.
- The only hypothesis which leads to a correct total length of the bar is H^{l1} with a $Conf_l(j_3,H^{l1})$ coefficient equal to 0.611 (the average of the $C_l^{(j_3,H^{l1})}(g)$ coefficients is equal to 5.5/6, and there are $N(j_3,H^{l1})=6$ groups of notes for $N'(j_3,H^{l1})=18$ notes). This score is lower than $Conf_l(j_2,H^{l0})$ for two reasons: the possibility degree of the third group is lower, and the false interpretation of symbol 4, classified as a note, breaks the group of eight notes.
- The total length of the bar is correct: $D(j_3,H^{l1})=4$ *beats* and the global score is equal to $Conf(j_3,H^{l1})=0.665*0.611=0.406$. This hypothesis can be discarded because this score is inferior to the score of the (j_2,H^{l0}) configuration.

Fig. 25 shows the results provided by the free demonstration software SmartScore [19] for Windows. There are three errors while the result provided by our program is completely correct. It is interesting to notice that this false configuration was included in the set of hypotheses evaluated by our program, but that it has not be chosen for two main reasons: first, because the compatibility coefficients were unfavorable to the flats, second because the uncertain duration dot B has been finally retained because it allows to satisfy the metric rule with a high possibility degree.

9. Results

All the algorithms described in Rossant [18] and in this paper have been implemented so that we are able to evaluate them by computing recognition rates, and to compare our program with a commercial software.

9.1. Experimental conditions

The program has been tested on 65 music sheets, which represents more than 25 000 symbols. Care is taken not to train on specific cases, by including in the test base music sheets providing from various composers and various publishers, consequently printed with different fonts, but all using the classical music writing conventions. The base includes examples of various levels of difficulty in terms of symbol density, rhythmical complexity and printing quality. The scanning has been performed on two different materials with no specific optimization of the binarization threshold, and the program has been running also without any parameters tuning, such as symbol model or threshold changes.

9.2. Recognition rates

Typical run-time is around 1.5 min on a pentium 1 GHz, without any correlation computing optimization or other implementation optimization. The average recognition rate of our program is around 98.4%. The errors are split up like this: confusions (0.3%), symbols missing (0.9%), non-existing symbols added (0.4%). There are also 1.65% of filled notes (quarter, eighth, sixteenth ... notes) which do not get the correct length.

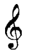
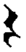
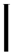
We will now detail these results for each class k. All the indicated rates will be from now on referred to the number of real symbols that have to be recognized, so without counting the non existing symbols output by our program. Tables 11 and 12 indicate the rates of symbols missing and added, referred to the number of occurrences belonging to the class ($r_k(k)$, Eq. (14)) and to the total number of occurrences ($r(k)$, Eq. (15)).

$$r_k(k) = \frac{\text{number of occurrences missing (added) in class } k}{\text{total number of occurrences belonging to class } k} * 100, \tag{14}$$

$$r(k) = \frac{\text{number of occurrences missing (added) in class } k}{\text{total number of occurrences}} * 100. \tag{15}$$

Two main reasons can be invoked to explain that some symbols are missing or added. The first one is simply the non detection. For example, when the vertical segment featuring many symbols is broken, the segmentation process fails. It is the most common defect which concerns essentially the accidentals but also some notes and bar lines. The second reason concerns especially the rests and is rather relevant to the decision algorithm. Indeed, this algorithm selects in priority the hypothesis groupings achieving the bar length constraint. So, when a bar does not have to achieve the metric constraint, for example in case of pick-up measures (Fig. 26a) that are not yet handled by our program, the algorithm may add silences or sometimes half-notes, and suppress others in order to reach the number of beats indicated by the time signature. It is the same phenomenon when the

Table 11
Rates of symbols missing, per class

Class k		$r_k\ (k)$	$r(k)$
0		0.00	0.00
1		0.58	0.02
2		2.06	0.05
3		2.16	0.11
4		8.95	0.07
5		8.36	0.32
6		7.37	0.03
7		7.05	0.10
8		0.00	0.00
9		1.02	0.01
10		5.05	0.02
11		0.00	0.00
12		0.04	0.03
13		1.18	0.01
14		0.00	0.00
15		1.12	0.11

correct solution is not in the set of hypotheses, for example when a bar line, a note or a dot is not detected (Fig. 26b), or in case of a misinterpretation of a note length (Fig. 26c).

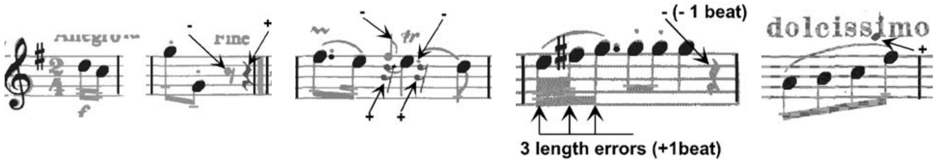
The rate of duration dots missing or added is rather important. It is indeed difficult to differentiate a dot from noise, and a significant number of them are outside of the defined search area, consequently not detected. But as shown later in Section 9.5, the use of the context helps efficiently to their recognition.

Lastly, another important result is the adding of grace notes generally resulting from the confusion with a text writing (Fig. 26d).

The confusion rate (0.32%) measures the rate of symbols which are misclassified, for example a natural classified as a sharp. It is rather low, indicating that the music writing rules have been correctly modeled. Table 13 details the results per class, indicating also the symbols they are mainly confused with.

Table 12
Rates of symbols added, per class

Class k		$r_k\ (k)$	$r(k)$
0	𝄞	0.00	0.00
1	♭	0.58	0.02
2	♮	0.32	$<10^{-2}$
3	♯	0.08	$<10^{-2}$
4	♩	7.46	0.06
5	•	2.04	0.08
6	𝄿	9.47	0.03
7	𝄾	0.54	$<10^{-2}$
8	𝄽	28.57	0.02
9	𝄽	6.48	0.08
10	𝄻	8.08	0.03
11	𝄼	12.50	0.02
12	●	0.06	0.04
13	𝅗	1.57	0.02
14	𝅝	0.00	0.00
15	|	0.00	0.00



(a) Pick-up measures (b) duration dot not detected (c) Correct length hypothesis missing (d) grace note adding

Fig. 26. Examples of symbols missing or added.

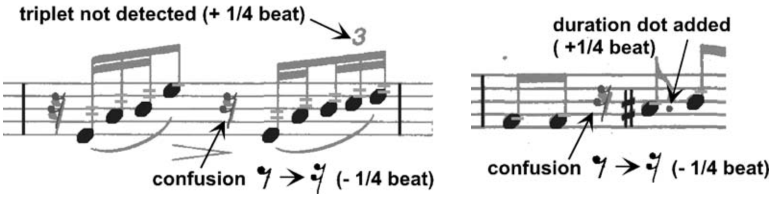
Table 13
Rates of confusions, per class

Class k		r_k (k)	$r(k)$	Mainly confused with
0	𝄞	0.00	0.00	
1	♭	1.16	0.04	♮
2	♮	1.43	0.03	♯, ♭, ♩
3	♯	0.69	0.04	♮
4	♩	8.96	0.07	♭, ♯, ♮
5	•	0.41	0.02	
6	𝄿	2.10	$<10^{-2}$	𝄾
7	𝄾	1.63	0.02	𝄽 𝄿
8	𝄾	0.00	0.00	
9	𝄽	0.34	$< 10^{-2}$	
10	▬	0.00	0.00	
11	▬	0.00	0.00	
12	𝅘	0.03	0.02	♩
13	𝅗	3.94	0.04	𝅘 ▬
14	𝅝	0.00	0.00	
15	|	0.23	0.02	𝅘

Most of the confusions between rests are again due to the bar length constraint. Fig. 27 shows how the algorithm chooses a bad rest in order to compensate a length error or a duration dot error. The confusion between accidentals is discussed in the next subsection.

9.3. Accidentals

It is interesting to study particularly the sub-group of the grace notes and the accidentals, only those that appear before a note, so without taking into account the accidentals indicating the tonality for which no confusion is possible. This time, we can consider that the length constraint does not have any influence. Thanks to the fuzzy modeling of the symbol classes and of the accidental use, the recognition rate has been increased by 3.2% and is now equal to 94.1%. The remaining errors are split up like this: confusions between accidentals (1.5%), confusion with other symbols (0.9%),


triplet not detected (+ 1/4 beat)
3
confusion 𝄾 → 𝄿 (- 1/4 beat)
duration dot added
(+1/4 beat)
confusion 𝄾 → 𝄿 (- 1/4 beat)

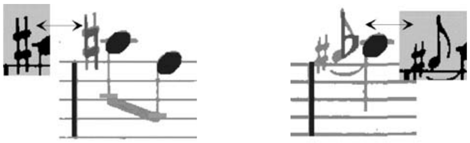
(a) confusion because of note length errors

(b) confusion because of the duration dot adding

Fig. 27. Examples of confusions.

Table 14
Confusion between accidentals (tonality excepted)

	♭	♮	♯	♩	—	Else
♭	**94.7**	3.0	0.00	0.00	1.9	0.4
♮	0.2	**96.5**	0.9	0.1	2.1	0.2
♯	0.0	0.8	**95.1**	0.00	3.7	0.4
♩	2.0	0.5	1.0	**82.1**	8.9	5.5



(a) confusion sharp/natural (b) confusion grace note/bemol

Fig. 28. Confusion between accidentals (tonality excepted).

accidentals missing (3.5%). Table 14 presents the confusion matrix C, defined as

$$C(i,j) = 100 * \frac{\text{number of symbols of class } i \text{ classified in class } j}{\text{number of symbols of class } i}. \qquad (16)$$

The two last columns are the rates of symbols missing or confused with a class not belonging to the sub-group of accidentals and grace notes.

The differences between the classes can be easily interpreted. We can first notice that there is no more confusion between flat and sharp: the graphical compatibility coefficient and the syntactic coefficient decide between them efficiently. There are still some confusions natural/sharp and natural/flat for two reasons: there are more cases of syntactic ambiguities (recall of an accidental), and the two symbols may be in some cases perfectly superimposed (Fig. 28a) so that the graphical

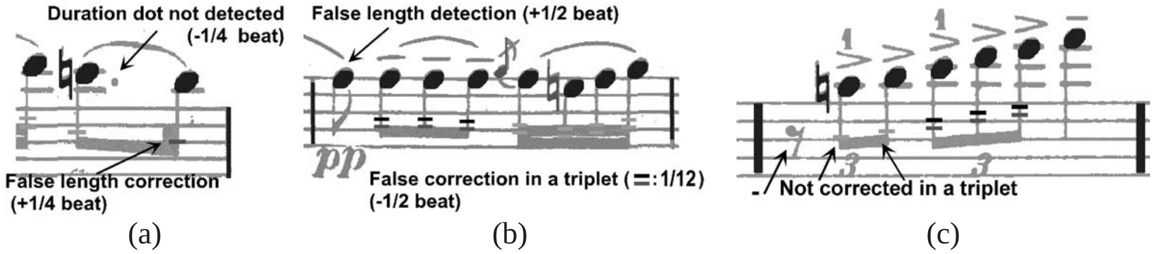

Duration dot not detected
(-1/4 beat)
False length correction
(+1/4 beat)
False length detection (+1/2 beat)
pp
False correction in a triplet (=:1/12)
(-1/2 beat)
Not corrected in a triplet
(a)
(b)
(c)

Fig. 29. Note length errors.

compatibility coefficient does not help to find the correct decision. Lastly, flats and grace notes may be very difficult to differentiate as shown in Fig. 28b.

9.4. Note length

1.65% of the filled notes (quarter, eighth, sixteenth … notes) get a wrong length. Some of the errors are again due to the non detection of a symbol (Fig. 29a), others compensate another length error (Fig. 29b). But the insufficient modeling of the different rhythmical configurations must be also invoked: all the possible note groupings, for example thus which associate triplets and sixteenth notes (Fig. 27a) are not yet handled by our program; and a rest cannot yet be associated with a group of beamed notes so that both rest and note length may be false (Fig. 29c).

9.5. Correction rates

In this Section we try to quantify the contribution of our fuzzy model by computing several rates for four main groups of classes: accidentals (tonality and grace notes included), notes, rests, duration dots (Table 15). A symbol is counted as initially correct when the decision process is right to choose the assumption of level H1 (column 1), and as a rightly corrected symbol (column 2) when the decision process is right to choose an assumption H2 or H3. 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 (column 5). Five corresponding rates are computed by $ri = ni/\sum_{j=1}^{5} ni$, where ni is the number of symbols belonging to column i. The expression $r = (n1 + n2)/\sum_{j=1}^{5}(nj)$ represents the average recognition rate.

This table can be compared with the results presented in Rossant [18]. It shows that the accidental recognition rate has been increased by 3.6%, although the tested music sheets present now more various printing fonts: more symbols are rightly corrected by the decision process thanks to the introduction of the fuzzy model of the symbol classes and of the relevant musical rules. The recognition rates of the rests and duration dots have been also improved, approximately by 8%. This is mainly due to the modeling of the common note groupings which has led to a decrease of the number of note length errors: now we get 98.35% of right length with a gain of correction of 3.6%. But these results can be still improved. The main research axes are the followings: improve the detection of the duration dots by expanding the search area and defining a compatibility degree

Table 15
Recognition rates per class

Initially right	Right corrected	Non-corrected	False corrected	Missing
Notes				
$r1$	$r2$	$r3$	$r4$	$r5$
99.38	0.47	0.02	0.07	0.06
$r = 99.85\%$				
Rests				
$r1$	$r2$	$r3$	$r4$	$r5$
73.87	20.64	0.22	0.77	4.50
$r = 94.51\%$				
Accidentals (tonality included)				
$r1$	$r2$	$r3$	$r4$	$r5$
88.24	8.05	0.40	1.13	2.17
$r = 96.29\%$				
Duration dots				
$r1$	$r2$	$r3$	$r4$	$r5$
78.49	12.74	0.00	0.41	8.36
$r = 91.23\%$				

with the dotted note; extend the modeling of note groups, and introduce associations of rests and note groups; introduce the pick-up measures, the dotted rest, the 1/32 rest.

9.6. Comparison with a commercial software

Although advertisements for commercial optical music recognition package claim very good recognition rates, the users agree that they are in practice too error-prone to be of much practical use. So it is interesting to verify if our work may be a contribution to improve this situation. In this section, we present on a whole music sheet a comparison between the results provided by our program and those provided by SmartScore 2.0 Professional Version [19]. This software is one of the most well known for Windows, and it is freely available for demonstration on the web. The last version allows now to enter some optional parameters: we have indicated that the number of voices is limited to 1, and we have allowed the recognition of triplets. Fig. 30a shows the original music sheet, Fig. 30b shows the results provided by SmartScore (left) and by our program (right). It is obvious that our fuzzy model integrating musical rules is able to solve some problems for which SmartScore fails. Regardless of the non-recognition of grace notes, there are with SmartScore 12 confusions between grace note and flat, natural and flat, sharp and flat, while there is zero confusion with our program; there are also respectively four and three accidentals missing. With SmartScore, 50% of the bars get a false length due to a duration dot missing or added (bars 1,6,14,19), confusions (bars 12,18), a note or a rest missing (bars 2,5,14), or note length errors (bar 3,4,18), and the analysis of these errors seems to prove that no model of the musical rules has been integrated in SmartScore recognition process. On the contrary, our program uses the knowledge of the common note groupings and of the bar length to arbitrate efficiently between several configurations. Indeed, even if the non recognition of a 1/32 rest, that is not yet handled by our program, is responsible for 2 other mistakes (bar 14), there are fewer errors in this bar than with SmartScore; moreover, the recognition is for the other bars perfect (two accidentals missing excepted) so that the global recognition is much more satisfactory.

9.7. Discussion

In this section we provide a discussion about the limitations of the proposed approach, related to the hypotheses we made, and about possible extensions.

The first hypothesis concerns the global information given as input: clef, key-signature, time signature. This is not a strong limitation since the user is ready to provide such an information, which guarantees good recognition rates, since the method thus uses very reliable information. This is typically the type of interaction which is acceptable, and allows to reach a good robustness. However, this assumption could be relaxed by introducing at the beginning of the process a recognition step dedicated to this global information. A method based on pattern matching similar to the one we use in the first part of the method is likely to perform well.

The second assumption we made is about monophonic music. This one is much stronger than the first one and cannot be easily relaxed. Possible extensions depend on the type of music. For orchestral scores, our approach could be extended quite easily, since they can be considered as a juxtaposition of monophonic lines. Each part could be analyzed individually using our method. The vertical alignment of symbols played at the same time can bring valuable additional information. Indeed, it can help in the decision step for choosing the appropriate length configuration in a bar. Rules expressing the relationships between parts could be added at this step. Therefore we can even expect better results for such scores, since the vertical alignment will confirm or infirm possible recognition hypotheses, and help in correcting inconsistencies, as in Blostein et al. [6] or in Coüasnon and Rétif [9].

For piano scores, or other scores of that type, the problem appears to be much more difficult to solve. The first step of the program, especially the pattern matching process can be adapted [18]. Chords for example can be easily treated by searching for note heads in a larger area along the note stem. Then an additional step is required to make the segmentation of the score into separate voices. Additional rules can be used for that [8]: vertical position criterion, direction of note stems, beamed notes. But this is a difficult step which raises a lot of problems, as illustrated in Fig. 31.

For instance, the number of voices is not necessarily constant in a bar, since some voices can merge (Fig. 31a), a voice is not restricted to stay on the same staff (Fig. 31b), etc. Rules dealing with such situations are probably more difficult to design than the ones concerning vertical alignments. After the voice separation, each voice could be again treated individually using our method. The global decision could also be made by adding rules checking the vertical alignment and length consistency, as suggested for orchestral scores. When this consistency cannot be achieved, it may be possible to go back to the voice separation step and check if other separation hypotheses could lead to a better consistency.

So our method could be extended to complex polyphonic music scores, unlike some other methods which are also designed for restricted musical configurations but seem to be very difficult to scale up.

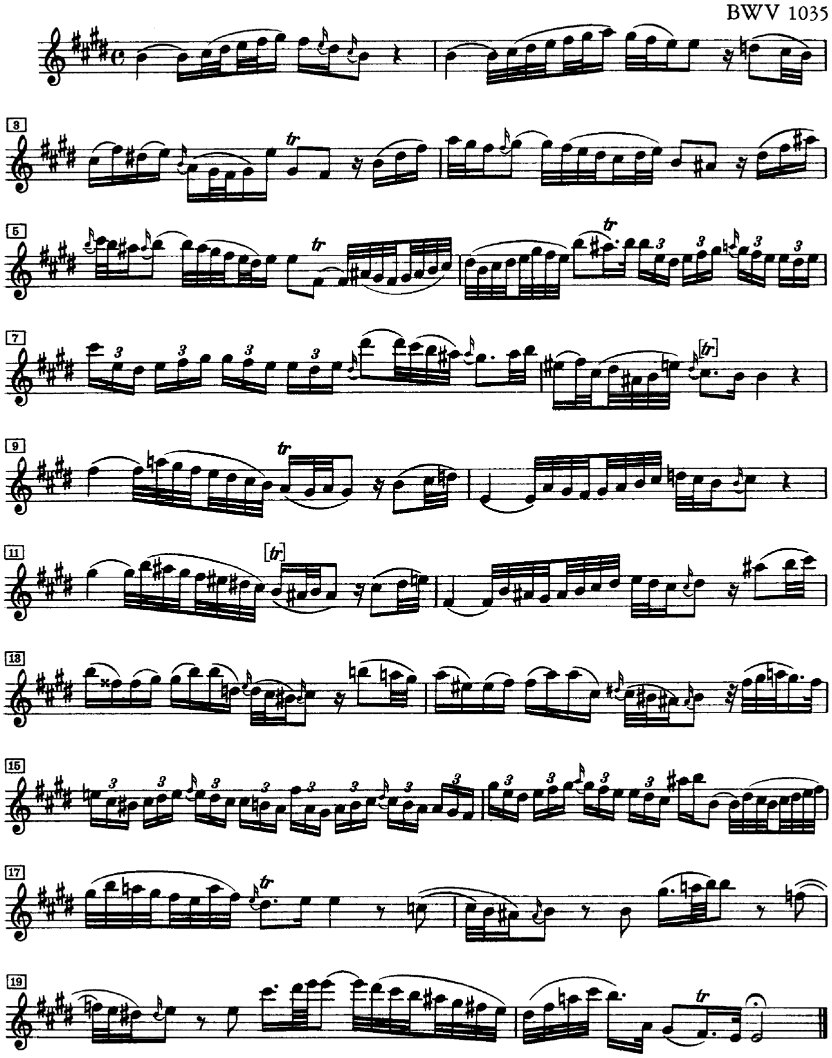


Fig. 30. (a) Original music sheet.

For example, Kopec et al. [14] use Hidden Markov Models to model uncertainty and obtain excellent results on very noisy images. But they restrict the application field to single-voice scores because the generative model is based on a finite state string grammar which is not adapted to represent distinct but mutually constrained event trains. Our results would certainly be more reliable in case of orchestral scores, but the complexity would be increased in case of piano scores. However we can expect that the number of configurations to process remains reasonable, because we can treat each voice individually before merging the results in the global decision step: an ambiguity concerning one particular symbol (for example between a natural and a sharp, or between a eighth rest or a sixteenth rest) will generally not affect the voice separation and consequently not affect the analysis of the other voices.

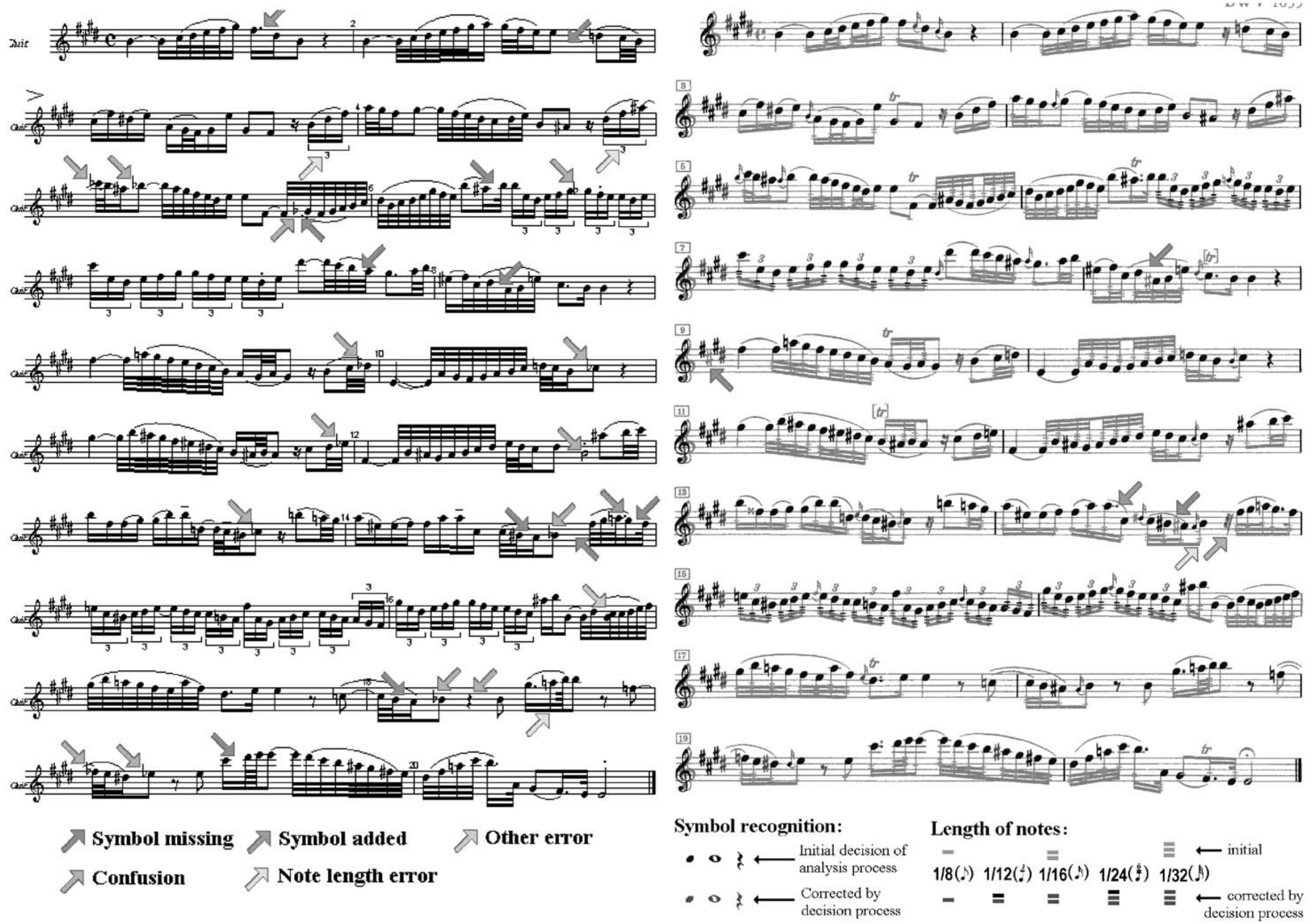


Fig. 30. (b) comparison between our program and SmartScore Professional Version.

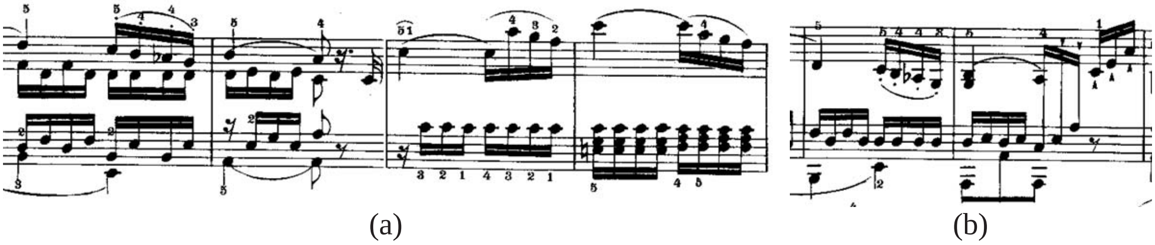


Fig. 31. Pieces of music extracted from a piano score.

10. Conclusion

In this paper, we proposed a fuzzy model for optical music recognition, integrating music writing rules. This model is introduced in the system after a first step which detects and analyzes separately the symbols [18]. A maximum of three recognition hypotheses are output for each of them, a recognition hypothesis assigning a symbol to a class. Then, the fuzzy modeling step presented in this paper is divided into three main parts: computing for each hypothesis a possibility degree of membership to a class through possibility distributions learned from the first analysis; computing compatibility degrees expressing graphical relationships between successive symbols according to their class; computing compatibility degrees expressing syntactic consistency between the symbols, according to their class and/or their length. The main musical rules related to the use of the accidentals and the metric have been expressed in this way. All these results are then merged into a single coefficient which must reflect the global consistency of the evaluated set of hypotheses according to all the criteria. This is done bar per bar, and the decision process simply chooses for each of them the configuration maximizing this global coefficient.

Our work is a contribution to the problem of the high level interpretation in optical music recognition. The proposed method is well suited to this problem for three main reasons: first, fuzzy modeling allows to express constraints which are more or less strict, which is essential to this application field. Indeed, most of the musical rules are just commonly but not mandatory used (for example the recall of an accidental, the common note groupings), or applied in an approximated way (relative position of the symbols). Secondly, fuzzy models allow to merge different kinds of constraints, here graphical and syntactical rules. Thirdly, the proposed method processes interactions between distant symbols, not only local interaction between successive symbols. Compared to an important commercial software SmartScore [19], our method has proven to be an interesting contribution, providing much more satisfactory interpretations. Unlike SmartScore, it is obviously able to reject strong misinterpretations thanks to our fuzzy modeling of the musical rules and its integration in a global decision process.

Three major improvements can be made in order to increase the recognition rate. The first one is to extend the reference base of note groupings, for example to the associations of triplets and eighth notes, and to include the possible associations of notes and rests. The second one concerns the detection of the duration dots: we must define a larger search area and define a compatibility degree between the dotted note and the dot, as a function of their relative position. The third one is to introduce some configurations that are not yet handled by our program: the dotted rests, the pick-up measures, the 1/32 rests. All these improvements will decrease the rate of length errors, and consequently the rates of confusion and the rates of added or missing rests.

Lastly, it could be interesting to extend our method to polyphonic music. This would require to allow the detection of several symbols at nearly identical horizontal locations and to re-arrange them into several voices. We can assume that a fuzzy model would be again an interesting way to express the common writing rules regarding the vertical alignment of the symbols and the length consistency between voices.

Acknowledgements

The authors would like to thank Evan Rothstein for his help in clarifying the terminology.

References

[1] J.-P. Armand, Musical score recognition: a hierarchical and recursive approach, in: 2nd Internat. Conf. on Document Analysis and Recognition, 1993, pp. 906–909.

[2] I. Bloch, Information combination operators for data fusion: a comparative review with classification, IEEE Trans. Systems Man Cybernet. 26 (1) (1996) 52–67.

[3] I. Bloch, Fusion of numerical and structural image information in medical imaging in the framework of fuzzy sets, in: P. Szczepaniak, et al., (Eds.), Fuzzy Systems in Medicine, Series Studies in Fuzziness and Soft Computing, Springer, Berlin, 2000, pp. 429–447.

[4] I. Bloch, H. Maître, Fusion of image information under imprecision, in: B. Bouchon-Meunier (Ed.), Aggregation and Fusion of Imperfect Information, Series Studies in Fuzziness, Physica Verlag, Springer, 1997, pp. 189–213.

[5] D. Blostein, H. Baird, A critical survey of music image analysis, in: H.S. Baird, et al., (Eds.), Structured Document Image Analysis, Springer, Berlin, 1992, pp. 405–434.

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

[7] N. Carter, R. Bacon, T. Messenger, The acquisition, representation and reconstruction of printed music by computer: a review, Comput. Humanities 22 (1988) 117–136.

[8] B. Coüasnon, J. Camillerapp, Using grammars to segment and recognize music scores, Internat. Assoc. for Pattern Recognition Workshop on Document Analysis Systems, Kaiserslautern, Germany, 1994, pp. 15–27.

[9] B. Coüasnon, B. Rétif, Using a grammar for a reliable full score recognition system, in: Internat. Comput. Music Conf., Banff, Canada, 1995, pp. 187–194.

[10] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, 1980.

[11] D. Dubois, H. Prade, R. Yager, Merging fuzzy information, in: J.C. Bezdek, D. Dubois, H. Prade (Eds.), Handbook of Fuzzy Sets Series, Approximate Reasoning and Information Systems, Kluwer, Dordrecht, 1999 (Chapter 6).

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

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

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

[15] R. Krishnapuram, J.M. Keller, Fuzzy set theoretic approach to computer vision: an overview, IEEE Internat. Conf. Fuzzy Systems, San Diego, CA, 1992, pp. 135–142.

[16] B. Modayur, Music score recognition—a selective attention approach using mathematical morphology, Technical report, Electrical Engineering Department, University of Washington, Seattle, 1996.

[17] K.C. Ng, R.D. Boyle, D. Cooper, Low and high level approaches to optical music score recognition, in: IEEE Colloq. on Document Image Processing and Multimedia Environment, 1995, pp. 3/1–3/6.

[18] F. Rossant, A global method for music symbol recognition in typeset music sheets, Pattern Recognition Lett. 23 (10) (2002) 1129–1141.

[19] SmartScore 2.0 Professional Version demo for Win9x/2000/NT/ME/XP, http://www.musitek.com/demopage.html.

[20] M.V. Stückelberg, D. Doermann, On musical score recognition using probabilistic reasoning, in: ICDAR, Bangalore, India, 1999, pp. 115–118.

[21] M.V. Stückelberg, C. Pellegrini, M. Hilaro, An architecture for musical score recognition using high-level domain knowledge, in: 4th Internat. Conf. on Document Analysis and Recognition, Vol. 2, 1997, pp. 813–818.

[22] M.-C. Su, C.-Y. Tew, H.-H. Chen, Musical symbol recognition using SOM-based fuzzy systems, in: Internat. Fuzzy System Assoc. Conf., Vol. 4, 2001, pp. 2150–2153.

[23] G. Watkins, The use of fuzzy graph grammar for recognising noisy two-dimensional images, in: North American Fuzzy Inform. Process. Soc. Conf., 1996, pp. 415–419.