USING A GRAMMAR FOR A RELIABLE FULL SCORE RECOGNITION SYSTEM[1]

Bertrand COÜASNON Bernard RÉTIF[2]
IRISA / INSA-Département Informatique
20, Avenue des buttes de Coësmes
F-35043 Rennes Cedex, France
couasnon@irisa.fr

ABSTRACT: Optical Music Recognition needs to be reliable to avoid users to detect and correct errors by controlling all the recognized score. Reliability can be reach by improving the recognition quality (on segmentation problems) and by making the system able to detect itself its recognition errors. This is possible only by using as much as possible the musical knowledge.

Therefore, we propose a grammar to formalize the musical knowledge on full scores with polyphonic staves. We then show how this grammar can help detection of most of errors on note duration. The presented system is in an implementation phase but is already able to deal with full scores and to point on errors.

1 Introduction

In structured document analysis, one important aim is to build a reliable system. It is reachable by improving the recognition quality (on segmentation problems) and by making the system able to detect itself its recognition errors. In Optical Music Recognition, full scores with polyphonic staves are difficult to recognize because of the density of musical information. Indeed, this density creates segmentation[3] problems such as two touching objects when syntactically they should not (*e.g.* accidental touching a notehead or an other accidental). These segmentation problems make recognition really difficult, because to recognize two connected objects you first have to segment (separate) them correctly; but to segment them correctly, you should have already recognized them! Current systems cannot solve this kind of problems [Blostein et al.92], but it is possible to solve most of them by using as much as possible knowledge on music writing.

Even if the segmentation is improved, a recognition system cannot be completely perfect. Indeed, the original document can be too damaged to be recognizable (*e.g.* a stem can be completely removed). A second way to make the system reliable is to give it the possibility to detect areas where it makes errors. The system can then point out these areas to the user for a "manual" correction. This avoids users from re-reading and checking the whole score, a work which can be very tedious. The user will only have to check areas pointed out by the system. It is possible to do this error detection, once again by using as much as possible context (musical knowledge).

All the existing optical music recognition systems use some musical rules, but those rules are usually selectively incorporated in the recognition process, and are not really formalized. However, Fujinaga [Fujinaga88] states that music notation grammar is context-free and LL(k). Hence, it is possible to define a grammar to formalize the music writing rules which can be found in complex scores. It is possible but difficult, indeed all the already proposed grammars are not able to deal with full scores.

We propose a method using a grammar, with new operators to define the relative object positions on a score. This grammar is separated from the program. Thus, it can then be modified and adapted in an easy way. It controls the whole recognition process improving recognition and detecting errors in order to get a system as much reliable as possible.

[1]In Proc. of the International Computer Music Conference, Banff, Canada, September 1995.
[2]Musician, in charge of computer music lectures at the University of Rennes II.
[3]To partition an image.

2 Related Work

In order to recognize musical scores, it is necessary to detect the staff lines (otherwise objects which are disconnected logically would appear as one single object). These can then be erased so that an initial segmentation process can be carried out on the objects (a technique used by most authors). The staff lines can be detected using different techniques, all of which treat the staff line as a real line. Only Carter [Carter et al.92] proposes a line method using a line-adjacency graph capable of dealing with staff lines that are not perfectly rectilinear (a relatively frequent case).

Once the objects have been segmented by erasing the staff lines, they are classified using different techniques (some authors use more than one): they are classified according to the bounding box size [Prerau75]; horizontal and vertical projections [Fujinaga88]; extraction of primitives such as notehead, stem, etc., using erosion, thinning or other techniques. Generally the label given to these primitives is not changed during the recognition process except in [Kato et al.92].

The main problems which then remain unresolved are [Blostein et al.92, Carter et al.92]:

- the scaling up of a prototype. Most of the systems which have been developed to date are prototypes adapted to simple scores. It is difficult to scale prototypes up to complex score systems (the same techniques cannot be used);
- the reconstruction of broken objects (noise);
- objects which touch when they should not.

The problems remain unresolved partly because a priori knowledge on the musical notation is not used to a sufficient extent. Although all the current systems for optical music recognition use some musical rules, these are usually selectively incorporated into the recognition process, and are not really formalized. Certain solutions do use some rules formalized by a grammar, but mostly for limited purposes (verification, error correction or final pitch calculation). They do not include graphical rules and are limited to simple scores.

Only Andronico [Andronico et al.82] proposes the use of a grammatical formalization, which includes the graphical level. However, the grammars are only adapted to simple and monophonic scores. As mentioned previously, Fujinaga [Fujinaga88] states that the grammar for music notation is context-free and LL(k). Hence, it is possible to define a grammar to formalize the music writing rules which govern complex scores.

3 Aims

Our objective is to design a system:

- capable of recognizing full scores (orchestra scores) where each instrument has its own staff, and the staves are vertically connected;
- able to deal with complex scores, usually polyphonic (different voices on a single staff);
- that is reliable enough to avoid the need for human verification (using the redundancies in the musical notation);
- that can be extended to handwritten scores (if they are not too badly written);
- that uses a vocabulary which can be extended relatively easily;
- capable of recognizing only "classical" music notation. The notation used by some composers in the 20^{th} century can differ considerably from one work to another.

4 Difficulties

Touching Objects The scores to be recognized are complex and therefore have a high density of symbols. As a result, connections are often found between musical objects that syntactically should not be touching, causing some segmentation problems (Fig. 1).

Broken Objects Scanning produces some noises. In addition, initial documents may be of poor quality, giving broken objects. Usually, the objects affected in this way are small (*e.g.* tuplets's figures). Staff lines must first be removed for the recognition process but this can also produce broken objects (this is the most frequent case).

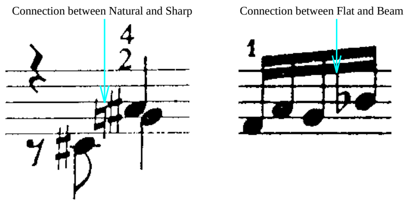

Figure 1: *Examples of objects that should not touch*

These segmentation problems are unavoidable. As traditional segmentation techniques at image level cannot solve the problems, we must rely as far as possible on the musical context.

5 Presentation of the Grammar

The objects on a score can be divided into two categories:

- constructs: (*e.g.* simple stemmed notes, beamed notes) composed of segments (such as stems, beams) and noteheads plus a set of construction rules which apply to these elements;
- symbolics: for instance clefs, accidentals, etc., these can be considered as characters. They can then be recognized using optical character recognition techniques (OCR).

Using this classification, grammar terminals are segments (part of a construct) and pixel arrays (for symbolics).To extract segments we use a technique founded on Kalman filtering [Pd et al.94]. For pixel array we start by computing connected components after staff line removing.

It is possible to distinguish between two levels of information on a musical score: a physical one corresponding to the way notes and their attributes (accidentals, accents, etc.) are formed are adjusted on the score; and a logical one corresponding to the syntactic way of using notes in written music. This means that the grammar also has to be structured on two levels: a graphical one corresponding to the physical level, and a syntactic one corresponding to the logical level. As the grammar describes an image, it is necessary to introduce special operators which define the relative position of the elements.

- Position Operator:

  A **P** B means A, and at the position **P** in relation to A, we find B.

- Factorization Notation (in association with the position operator):

  A (**P1** B : **P2** C) means A **P1** B and A **P2** C

- To define the extension of the rule for a single staff R to a rule for a system of staves :

  $map\_staff(R)$

To present the grammar, we will take some simple rules and describe graphically the grammar structure (Fig. 3)(Fig. 4). We can not present here all the complete grammar as it takes 5 pages of description.

5.1 Graphical Part

In order to illustrate the graphical level (Fig. 3) of the grammar, we present (Fig. 2) a set of simplified rules which defines a beamed eighth note. There is only one beam that links the notes. The chords, dots, accents and slurs are not defined in this example:

The grammar variables represent:

consRest	eighth rest, sixteenth rest... (constructed rests)
noteGr	single graphical note (half, quarter, eighth...)
noteGrU	graphical note with an upwards-oriented stem
noteGrD	graphical note with a downwards-oriented stem
noteHead	notehead with all its attributes
head	can be a black head or a white head
accidental	can be natural, flat, sharp, double-flat, double-sharp

The position operators have the following meaning:

leftTip	at the left tip
closeV	vertically close
rightTip	at the right tip
closeLftDownTip	close on the left of the down tip
closeLftSameLine	close on the left on the same pitch

And the rules are:

beamedNote::= beam (**leftTip** noteGr :
[**closeV** [noteGr | consRest]]* :
rightTip noteGr)
noteGr ::= noteGrU | noteGrD
noteGrU ::= stem **closeLftDownTip** noteHead
noteHead ::= head **closeLftSameLine** [accidental]$^?$

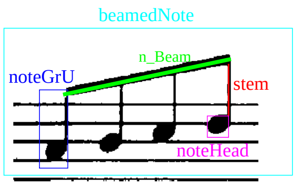

Figure 2: *Graphical representation of the rule beamedNote*

Consequently, we can interpret the rule noteGrU as follows:

> There is a noteGrU if there is a stem, and close at the left of the down tip of the stem, there is a noteHead

The noteGrD (graphical note with a downwards-oriented stem) is interpreted similarly.

The factorization notation (:) is illustrated by the rule beamedNote. This rule means that a beamed note is made of a beam and at each tip of this beam there is a graphical note and in between there is some graphical notes. Each position operator **leftTip**, **closeV** and **rightTip** refers to the rule stem. This is useful when different objects are positioned in relation to a single other one.

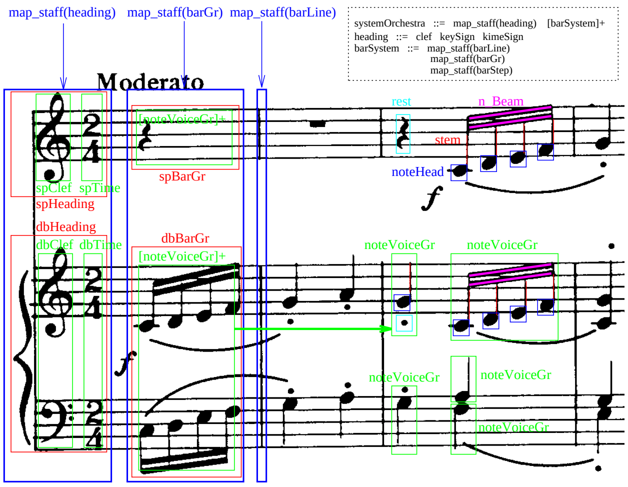

Figure 3: *Beginning of the grammar and map_staff(*barGr*) (graphical level)*

At the graphical level (*map_staff*(barGr))(Fig. 3), only notes and their attributes are recognized by using the rules on the way they are formed and adjusted on the score. There is no notion of polyphony in this part but only a notion of set of notes (noteVoiceGr). Polyphony is treated in the syntactic part.

5.2 Syntactic Part

Fig. 4 shows the syntactic level of the same piece of music as Fig. 3.The notion of Step creates a vertical cutting of the whole score. A Step corresponds to the smallest duration in a column of vertically aligned notes (in a system). This notion of step solves some of the problems due to polyphony and full scores. Indeed, it can manage the simultaneity (i.e. the verticality) of notes on a same staff (case of polyphony), and on the different staves of one system (full scores).

Informations such as dynamic markings, phrasing slurs... which are associated to the staff and not to a note, are introduced in the grammar at the Step level.

5.3 Staves System

We can illustrate the use of the *map_staff* operator by seeing the rule systBar:

systBar ::= *map_staff*(barLine)
map_staff(barGr)
map_staff(barStep)

For example, if the system have 4 staves, the operator *map_staff* will apply 4 times the rule barLine, then the second *map_staff* will apply 4 times the rule barGr to detect all the notes and 4 times the rule barStep to make the syntactical checking. All the attributes are automatically given to the right staff.

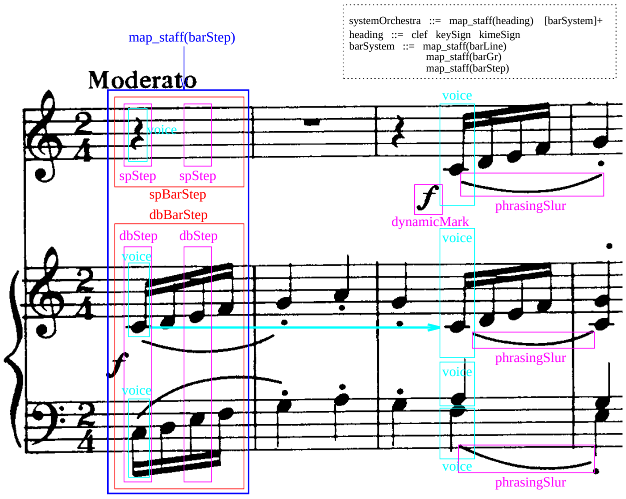

Figure 4: *map_staff(barStep) (syntactic level)*

6 Using the Grammar to Detect Note Duration Errors

A recognition system have to be reliable to be efficient and useful. Indeed, reliability avoids a "manual" checking of the recognized score. Thanks to musical notation redundancy, this can be done by making different controls to detect errors. This detection is possible if the musical context is fully used like in the grammatical formalization we proposed. Errors to detect can be dued to:

- the recognition system itself;
- the bad quality of documents. Indeed, scanned documents can have some defects, due most of time to a bad printing quality or a deteriorated document. These defects can create a stem vanishing or a black head changed in a white one. Some "salt and pepper" areas can make the recognition of a musical object impossible. Nevertheless, musicians usually can go over this bad quality with the help of the musical knowledge (context).

Excepted the graphical context (formalized in the grammar), musical context can be syntactic or semantic. As syntactic information, we have the number of beats in a bar according to the time signature. We can also use, in full scores, the vertical alignment of notes starting at the same time. Those informations are used in the syntactical part of the grammar. As semantic information, we could use rules of harmony, or rhythm patterns: when a pattern is found on a staff there is a great chance to find the same pattern on other instruments or in the next bar. However, those rules are too restrictive to be used.

We will only use the syntactic context which is general enough (even if some composers sometimes, for example, do not follow the number of beats rule). Moreover this context is able to detect (with the aligned notes) the same errors than those detected by the rhythm patterns.

Error detection on the number of beats in a bar can easily be done through grammatical attributes. These attributes are used to compute the number of beats and compare it with the time signature. Thanks to the *map_staff* operator, the "horizontal" attribute circulation is done automatically.

Inconsistency detection on note alignment-duration is more difficult. Indeed, the beat when a note starts have to be compared with all beats when aligned notes on other voices and staves start.

To do this, we use the notion of Step presented with the grammar. To be able to compare a note position (time and graphic) with the others, we propose a grid containing for each Step the aligned notes position and the Step duration. This Step duration is the smallest one found in starting or finishing notes at the Step position. Therefore, a Step contains at least one starting note and possibly continuing notes (notes started at a previous Step with a longer duration than the Step one). This grid produces a time cutting up of scores where an ideal note duration according to all the others is associated to each Step.

The grid is compute between *map_staff*(barGr) and *map_staff*(barStep) where all notes on all staves of a bar are detected. There is enough information in the grid to detect all time-alignment inconsistencies by simply compare the duration in the grid with the note duration. This avoid a direct comparison between a note and all the others in the bar. Moreover, it is possible to use a correction of a note duration when an error is detected.

In the future, it should be possible to improve the recognition by going back to the image level to check the presence of an unrecognized note detected through the grid.

7 Results

At present, a grammar has been defined for full scores, with different voices on a single staff, chords on a voice, accents, tuplets (*e.g.* triplet), pause, octave, dynamic markings, phrasing slurs, rhythmic slurs and appoggiatura. Abbreviations, ornaments and lyrics are not yet included.

Most of the tools needed by the parser and some of the position operators have already been defined. As the grammar and the parser are mutually independent, it is possible to test the parser using a small grammar without losing its generality. We have defined a subset of the initial grammar to recognize full polyphonic scores with clefs, key and time signature, half, quarter, single eighth notes and beamed notes, accidentals, dots after a note, rests, bar lines, and pitch processing. As the system can recognize full scores, it is also able to carry out two kinds of check: first on the number of beats in a bar (per voice) according to the time signature and second on the vertical alignment of notes in a system. These two checks enable the system to detect most errors and usually propose a method of correction. The grammatical formalization makes possible some improvement on segmentation problems (*e.g.* noteheads touching accidentals) [Couasnon et al.95b].

The parser [Couasnon et al.95a] is implemented in a compiled λProlog, a typed dialect Prolog integrating high-order unification. The choice of this language allowed us to produce an efficient prototype adapted to the complexity of the problem. Furthermore we have already written a grammar compiler which automatically produces the parser. At the end of the parsing, this system can point out unrecognized objects or bars which do not match the music syntax (which are likely to be poorly recognized).

As a result of the constraints on development time, we still have to incorporate the classification system for the symbols, and the segmentation and merging of connected components. The current classification system is very simple, but we are currently carrying out improvements. These elements are tools for the parser, and should not be a problem for scaling the system. The parser is actually the kernel of the system and is independent of the grammar and the classification system, and of the segmentation and merging tools. Once the parser works, it will be possible to scale up the system by simply changing the grammar given to the parser, while the tools will be defined separately.

8 Conclusion

In this paper we have presented a recognition system for music scores that is fully controlled by a grammar. The grammar which currently exists can cope with full scores, with different voices on a single staff, with chords on a voice, and is also capable of recognizing accents, phrasing slurs, dynamic markings, etc. Unlike other systems, our approach formalizes all the rules used for recognition and permits maximum integration of the context. The formalization also produces a system which is homogeneous.

In most other systems, the syntax is used merely to check a label. Here, on the other hand, the syntax controls the entire recognition process, and therefore produces a more reliable label. Usually, the grammatical methods are used only at a high level, whereas this system goes back to the image level to produce an accurate segmentation and thus accurate recognition.

One advantage of the formalization of the grammar is the separation between the operating part of the system and the definition of musical rules. This separation means that rules can be easily modified or the system can be adapted for another kind of structured document. In fact, a new recognition system, using the same parser, can be constructed simply by defining a new grammar.

Moreover, this system provides a solution to the problems which currently remain unresolved, as described by Blostein [Blostein et al.92] and Carter [Carter et al.92]: that is the reconstruction of broken objects, and the segmentation of touching objects. It should also help to solve the problem of scaling up the system. It is already able to deal with full scores, polyphony (for the moment only 2 voices per staff). With the help of the number of beats in a bar (according to the time signature) and the vertical alignment of notes (in full scores) the system can detect and correct recognition errors on note duration producing more reliable results.

References

[Andronico et al.82] Andronico (A.) and Ciampa (A.). – On automatic pattern recognition and acquisition of printed music. *In : ICMC, International Computer Music Conference*, pp. 245–278. – Venice, Italy, 1982.

[Blostein et al.92] Blostein (D.) and Baird (H.). – A critical survey of music image analysis. *In : Structured Document Image Analysis*, éd. par Springer-Verlag, pp. 405–434. – Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[Carter et al.92] Carter (N. P.) and Bacon (R. A.). – Automatic recognition of printed music. *In : Structured Document Image Analysis*, éd. par Springer-Verlag, pp. 456–465. – Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[Couasnon et al.95a] Couasnon (B.), Brisset (P.) and Stephan (I.). – Using logic programming languages for optical music recognition. *In : International Conference on the Practical Application of Prolog*, pp. 115–134. – Paris, France, April 1995.

[Couasnon et al.95b] Couasnon (B.) and Camillerapp (J.). – A way to separate knowledge from program in structured document analysis: Application to optical music recognition. *In : ICDAR, International Conference on Document Analysis and Recognition*. – Montreal, Canada, August 1995.

[Fujinaga88] Fujinaga (I.). – *Optical Music Recognition using projections*. – Montreal, Canada, Master's thesis, McGill University, Faculty of Music, 1988.

[Kato et al.92] Kato (H.) and Inokuchi (S.). – A recognition system for printed piano music using musical knowledge and constraints. *In : Structured Document Image Analysis*, éd. par Springer-Verlag, pp. 435–455. – Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[Pd et al.94] Poulain d'Andecy (V.), Camillerapp (J.) and Leplumey (I.). – Kalman filtering for segment detection: Application to music scores analysis. *In : ICPR, 12th International Conference on Pattern Recognition (IAPR)*, pp. 301–305. – Jerusalem, Israel, October 1994.

[Prerau75] Prerau (D. S.). – Do-re-mi: A program that recognizes music notation. *Computer and the Humanites*, vol. 9 (1), January 1975, pp. 25–29.