UTILISATION D'UNE GRAMMAIRE DANS LA RECONNAISSANCE DE PARTITIONS D'ORCHESTRE*

Bertrand COÜASNON † Bernard RÉTIF ‡

Résumé

La reconnaissance optique de partitions musicales nécessite d'être fiable afin que l'utilisateur n'ait pas à relire entièrement la partition produite par le système, pour détecter et corriger les erreurs.

Cette fiabilité peut s'obtenir, d'une part en améliorant la qualité de la reconnaissance, notamment sur les problèmes de segmentation, et d'autre part en faisant détecter par le système lui-même ses éventuelles erreurs de reconnaissance. Ces deux moyens peuvent être mis en place uniquement à condition d'utiliser au maximum la connaissance a priori sur l'écriture musicale.

Nous proposons donc une grammaire permettant de modéliser la connaissance sur l'écriture des partitions d'orchestre à portées polyphoniques. Nous indiquerons ensuite, comment avec l'aide de cette grammaire, il est possible de détecter une grande partie des erreurs sur la durée des notes. Le système présenté est en cours de développement, mais permet déjà de reconnaitre des partitions d'orchestre en indiquant les zones comportant des erreurs.

1 Introduction

Lorsque l'on s'intéresse à la reconnaissance de documents, l'objectif premier est de réaliser un système le plus fiable possible. Une première possibilité pour fiabiliser un système est d'améliorer ses capacités de reconnaissance afin d'analyser des documents plus complexes. Dans le cadre des partitions musicales, les partitions d'orchestre avec portées polyphoniques sont difficiles à reconnaître car l'information musicale est généralement dense. Or, cette densité provoque des problèmes de segmentation [1], liés par exemple à deux objets se touchant alors que, syntaxiquement, ils ne le devraient pas : altération touchant une tête ou une autre altération. Ces problèmes de segmentation rendent la reconnaissance de ces objets très difficile car pour reconnaître correctement deux objets connexes il faut les segmenter (les séparer) correctement ; mais pour les segmenter correctement il faudrait les avoir reconnus avant. Bien que les systèmes existants ne peuvent résoudre ce genre de problème [2, 4], il existe un moyen pour surmonter cette difficulté, en utilisant au maximum la connaissance a priori que l'on a sur l'écriture musicale.

Cependant, dans certaines conditions, tous les efforts possibles pour améliorer les capacités de reconnaissance du système seront insuffisants. En effet, le document d'origine peut parfois être trop dégradé pour être reconnu, ainsi par exemple une hampe peut être complètement effacée. Une seconde possibilité pour fiabiliser un système est de lui permettre de détecter ces zones où il fait des erreurs afin de les signaler pour réaliser une correction «manuelle» ultérieure. En effet, cette méthode a pour effet d'éviter à l'utilisateur de relire et vérifier l'ensemble de la partition, ce qui peut prendre énormément de temps [3]. Il reste alors uniquement à vérifier les zones détectées par le système. Cette détection est possible car dans une partition, l'information est redondante, comme par exemple l'alignement vertical des notes synchronisées. Mais pour réaliser cette détection d'erreur il faut à nouveau utiliser au maximum le contexte (*i.e.* la connaissance a priori).

Nous proposons donc une méthode fondée sur la syntaxe musicale pour obtenir un système le plus fiable possible. L'écriture musicale étant complexe à modéliser, il est nécessaire de la séparer du programme, afin de pouvoir l'adapter facilement. Nous utilisons donc une grammaire pour représenter cette

*. Dans les actes des Journées d'Informatique Musicale, Paris, France, avril 1995.

†. IRISA / INSA-Département Informatique, 20, Avenue des buttes de Coësmes, F-35043 RENNES Cedex, FRANCE. Email : couasnon@irisa.fr

‡. Musicien et chargé de cours à l'Université de Rennes II.

1. Réaliser des regroupements de pixels dans l'image, généralement pour isoler une information utile.

connaissance. Nous débuterons par l'étude des méthodes existantes, puis nous présenterons la grammaire (modélisant des partitions d'orchestre) qui constitue le coeur de notre système. Nous verrons ensuite comment utiliser cette grammaire pour détecter, voire corriger les erreurs temporelles. Nous terminerons en exposant les résultats que nous avons obtenu.

2 Méthodes existantes

La reconnaissance de partitions nécessite une phase de détection des lignes de portée sinon les objets logiquement séparés apparaitraient comme un seul objet. En effet, cette phase de détection permet, de réaliser une première segmentation des objets après suppression des lignes (technique utilisée par la plupart des auteurs), et surtout de pouvoir calculer par la suite la hauteur des notes.

Cette détection est réalisée à l'aide de différents traitements mais le plus souvent en considérant la ligne de portée comme une droite, ce qui permet d'utiliser des projections horizontales [8, 9] ou des recherches de cordes les plus longues [10, 12]. Seul Carter [4] propose une méthode à base du graphe des lignes adjacentes permettant de détecter des lignes de portée non-parfaitement rectilignes (cas assez fréquent).

Les objets, segmentés à la suite de la suppression des lignes de portée, sont ensuite classés selon différentes méthodes (certains auteurs en utilisent plusieurs) : pré-classification sur la taille des boites englobantes[11, 5] ; projections horizontales et verticales [8] ; extraction de primitives (avec des techniques d'érosion, de squelettisation…) comme tête de note, hampe…Il n'y a généralement pas de remise en cause de cet étiquetage [2] de primitives sauf pour Kato [9].

D'après Blostein [2] et Carter [4], les principaux problèmes restant à résoudre sont les suivants :

- la majorité des systèmes développés sont des prototypes adaptés à des partitions assez simples. Le passage à des partitions complexes pose des difficultés (les mêmes techniques ne pouvant plus être utilisées). Par exemple la difficulté de passer d'un système monophonique à un polyphonique.
- reconstituer un objet segmenté par le bruit
- le problème des objets connectés alors que, syntaxiquement, ils ne devraient pas l'être.

Le manque d'utilisation de la connaissance a priori est l'une des raisons pour lesquelles ces problèmes sont encore ouverts, bien que les différents systèmes de reconnaissance de partitions intègrent tous des règles de l'écriture musicale. Mais le plus souvent, ces règles sont réparties de manière ponctuelle au sein du processus de reconnaissance et manquent généralement de formalisation précise. Certaines solutions utilisent des règles formalisées par une grammaire mais elles sont alors principalement utilisées en vérification ou en correction d'erreurs et, éventuellement, en calcul des hauteurs finales des notes (variant avec les clés, altérations…). Cependant, elles n'intègrent pas les règles existant au niveau graphique et restent limitées à des partitions très simples.

Seule la solution d'Andronico [1] tend vers ce type de formalisation jusqu'au niveau graphique en utilisant une grammaire «naturelle» et un ensemble de grammaires «bidimensionnelles», mais ces grammaires n'étaient adaptées qu'à des partitions simples et monophoniques. Fujinaga [7] définit la grammaire de la notation musicale comme étant algébrique et LL(k), ce qui permet au musicien (analyseur descendant) de lire cette notation de façon efficace. On peut donc s'orienter vers la définition d'une grammaire formalisant les règles de l'écriture musicale (que l'on peut rencontrer sur les partitions complexes), contrôlant tout le processus de reconnaissance afin d'y utiliser au maximum le contexte. Afin d'éviter de futurs problèmes d'extension de prototype, nous allons fixer les objectifs et, en fonction de ceux-ci, déterminer le système de reconnaissance.

3 Objectifs

Nous voulons développer un système capable de :

- reconnaître les partitions d'orchestre,
- reconnaître les partitions complexes (souvent polyphoniques),

2. Associer un nom à un objet graphique

- avoir le moins possible d'erreurs de reconnaissance, et quand il y en a les détecter, afin d'éviter de vérifier «manuellement» l'ensemble de la partition,
- être adaptable aux partitions manuscrites,
- augmenter son vocabulaire relativement facilement,
- reconnaître uniquement la notation «classique», la notation en musique comptemporaine étant trop complexe.

Afin de définir une grammaire permettant de représenter une partition, nous allons dans un premier temps étudier le type d'informations que l'on peut y rencontrer.

4 Informations rencontrées sur une partition

Nous pouvons, comme l'a montré Martin [10], séparer en deux classes les objets se trouvant sur une partition :

- les construits : nommés «iconiques» par Martin (notes, groupes de notes, soufflets, ...) et constitués de segments (hampe, barre de groupe...) et éventuellement de têtes (de notes, de fraction de soupirs) ou de crochets, auxquels sont associés un ensemble de règles d'assemblage de ces segments et têtes permettant de décrire leurs diverses formes. Les barres de mesure, par leurs longueurs diverses entrent également dans cette classe.
- les symboliques : par exemple les clés, altérations, pauses... peuvent être considérés comme des caractères. Les techniques utilisées en reconnaissance de caractères pourront donc leur être appliquées.

Comme les objets construits possèdent des formes diverses mais suivent des règles précises, il est nécessaire pour les reconnaître que la grammaire définisse ces règles. La grammaire devra également pouvoir manipuler la seconde classe d'objets : les symboliques.

On peut constater sur une partition une indépendance entre la manière de regrouper les notes et la sémantique musicale (si l'on ne s'intéresse qu'à la reconnaissance et non à l'interprétation). En effet, on peut considérer un groupe de notes comme étant syntaxiquement complètement équivalent à une suite de croches simples. Or l'information de groupe est importante pour faciliter la reconnaissance, mais rend plus difficile la gestion de la simultanéité des notes dans une partition d'orchestre. Il nous faut donc distinguer deux types d'informations que l'on peut rencontrer sur une partition :

- les informations physiques : correspondant à la manière dont les notes et leurs attributs (altérations, signes d'attaques...) sont formées et agencées sur la partition.
- les informations logiques : correspondant à la manipulation des notes de manière syntaxique dans l'écriture musicale. Ces notes sont alors indépendantes des regroupements, et possèdent en attribut toutes les informations les concernant : hauteur, durée, altération... En effet, à ce niveau syntaxique, il n'est pas nécessaire de savoir, par exemple, où doit se trouver un dièse par rapport à la note, mais simplement que cette note est altérée par un dièse.

  Cette concentration de l'information musicale autour de la note permet entre autre au musicien de vérifier la synchronisation des notes entre les différentes portées d'une partition d'orchestre.

Cette séparation est également nécessaire pour reconnaître par exemple une clé au milieu d'un groupe de notes. En effet, cette clé ne peut être reconnue en même temps que le groupe car les modifications dues à la clé s'appliquent à toute la portée et non pas seulement au groupe.

La grammaire devra donc modéliser cette séparation de l'information musicale en deux niveaux.

5 Les terminaux

La grammaire doit permettre de décrire un document et de définir un contexte utilisable dans le processus de reconnaissance. Pour ce faire, les terminaux doivent suivre deux contraintes : d'une part, être les entités de base de la description du document et d'autre part, être reconnaissables sans information contextuelle, afin d'avoir une information sur laquelle le processus de reconnaissance puisse s'appuyer.

D'après les objets rencontrés sur une partition (les symboliques et les construits) on peut considérer le segment d'ensemble des symboles comme vocabulaire terminal. En effet, les construits peuvent être modélisés à l'aide du segment et de têtes ou crochets (des symboles). Les symboliques eux, sont naturellement représentés par des symboles.

De plus, l'information segment est suffisamment simple et locale pour être reconnue sans contexte. Les symboles, qui peuvent être vus comme des caractères, sont reconnaissables sans contexte en utilisant les techniques d'OCR (Optical Character Recognition), à condition qu'ils soient correctement segmentés. Cette condition est réalisée suffisamment souvent après la suppression des lignes de portées pour que le système puisse, dans le cas d'erreurs de segmentation, les corriger à l'aide du contexte.

6 Description de la grammaire

6.1 Structure

La structure de la grammaire dépend bien sûr de ce que l'on souhaite reconnaître. D'après les objectifs que nous nous sommes fixés, nous devons, entre autres, reconnaître des portées polyphoniques et réaliser de la détection d'erreurs. Celle-ci nécessite de contrôler par exemple le nombre de temps dans une mesure. Or cette vérification n'est possible que voix par voix. Il faut donc que le système soit capable de répartir correctement les notes dans les voix. En conséquence, la grammaire devra intégrer cette notion de voix. On aura donc jusqu'à 3 voix par portée. Mais, par exemple dans une partition pour piano, la portée double peut comporter une voix passant de la portée supérieure à la portée inférieure. Il est donc nécessaire de ne pas considérer ces doubles portées comme deux portées séparées, mais bien comme une seule portée comportant 11 lignes et autorisant jusqu'à 5 voix simultanées. La grammaire doit donc faire la distinction entre les portées simples (les non-terminaux débutant par les lettres «sp») et leurs équivalents pour les portées doubles (débutant par «db»).

Afin de simplifier le traitement, et de plus la vérification du nombre de temps dans une mesure se faisant au niveau de la mesure, la portée sera naturellement décomposée en mesures dans la grammaire.

Nous avons vu précédemment les deux niveaux physiques et logiques que l'on trouve sur une partition. Ces deux niveaux doivent se retrouver dans la grammaire. En effet, par exemple, pour vérifier le nombre de temps dans une mesure, il n'est pas nécessaire de connaître la forme d'une note, ni où sont placés ses attributs. Mais seulement qu'une note existe et possède des attributs. Nous allons donc séparer la reconnaissance en deux phases :

- une graphique (correspondant au niveau physique) qui reconnait les notes en décrivant leur forme et le positionnement relatif de leurs attributs (altérations, points, signes d'attaque).
- une syntaxique (correspondant au niveau logique) qui reconnait tous les symboles se rattachant à une voix comme les liaisons, nuances..., ou à une portée comme les clés, alta, bassa, nuances, liaisons...C'est également à ce niveau que les notes sont réparties voix par voix et que sont controlés le nombre de temps par voix par mesure et l'alignement vertical des notes au sein d'une portée polyphonique ou d'un système d'orchestre.

6.2 Syntaxe, opérateurs

Avant de pouvoir écrire la grammaire, il faut définir une syntaxe et des opérateurs adaptés aux particularités du domaine. En effet, la grammaire devant décrire une image, c'est à dire un espace à 2 dimensions, il est nécessaire d'introduire des opérateurs particuliers permettant de définir le positionnement relatif des éléments. Ces opérateurs permettant en outre de s'affranchir des problèmes de non connexité en manuscrit, où par exemple les hampes le plus souvent ne touchent pas les têtes de notes. Ces opérateurs de position possèdent la sémantique suivante :

$$A \mathbf{P} B$$

signifie A, et à la position $\mathbf{P}$ par rapport à A, on trouve B.

Cet opérateur permet de définir avec beaucoup de souplesse un lien entre 2 objets en spécifiant leurs positions relatives jusqu'à éventuellement, la zone de l'objet A en relation avec l'objet B. Cette souplesse offre la diversité des opérateurs et on trouvera dans la grammaire, pratiquement un opérateur différent par lien entre objets.

Nous allons utiliser pour la grammaire la syntaxe suivante :

::=	le constructeur de règle
en italique	les terminaux
en gras	les opérateurs de position
en standard	les non-terminaux

Nous ajoutons à cette syntaxe les abréviations classiques des expressions régulières :

A^*	répéter A de 0 à n fois
A^+	répéter A au moins une fois
$A^?$	répéter A 0 ou 1 fois
A^{n-p}	répéter A de n à p fois
$\mid$	ou

L'utilisation des opérateurs de position nécessite de faire référence à la variable grammaticale à gauche de l'opérateur pour positionner la variable grammaticale de droite. Mais, il peut être nécessaire qu'une même variable grammaticale serve de référence à plusieurs opérateurs de position afin de placer plusieurs autres variables grammaticales. Ceci est possible grâce à l'utilisation d'une notation de factorisation :

$$A(B : C)$$

signifie $A\ B$ et $A\ C$

Cette notation n'étant utilisée qu'en association avec des connecteurs de positionnement, on obtient plutôt :

$$A(\mathbf{P_1}\ B : \mathbf{P_2}\ C)$$

signifie $A\mathbf{P_1}B$ et $A\mathbf{P_2}C$

Parmi les objectifs, nous nous sommes fixés la reconnaissance de partitions d'orchestre. Afin de réaliser cet objectif le plus facilement possible, nous définissons un opérateur SYST permettant d'appliquer une règle adaptée à une portée simple, à l'ensemble des portées d'un système (c'est à dire à une liste de portées), générant ainsi de manière automatique la règle pour un système de portées. Cet opérateur SYST s'occupe bien évidemment de gérer de manière transparente la circulation des attributs qui par exemple doivent circuler «horizontalement» sur une même portée simple. La syntaxe de SYST est la suivante :

$$SYST(R)$$

pour définir l'extension d'une règle R pour une portée simple, à la règle adaptée au système complet.

Cet opérateur permet une extension simple de la grammaire pour portée simple vers une grammaire adaptée aux partitions d'orchestre.

Bien que la grammaire offre une approche descendante de l'écriture musicale, nous allons, pour une meilleure compréhension, réaliser la présentation de la grammaire de manière ascendante. Nous verrons ainsi en premier la partie graphique, puis syntaxique et enfin la vue globale de la grammaire. Nous ne pourrons présenter ici la grammaire complète car elle représente 5 pages de règles.

6.3 Présentation de la partie graphique

Afin d'illustrer le niveau graphique (voir figure 2) de la grammaire nous allons présenter (voir figure 1) un ensemble de règles très simplifiées définissant un groupe de croches. Ainsi, il n'y a qu'une seule barre de groupe reliant les notes. Les accords, les points d'allongement de durée d'une note, les signes d'attaque et les liaisons rythmiques ne sont pas définis dans cet exemple.

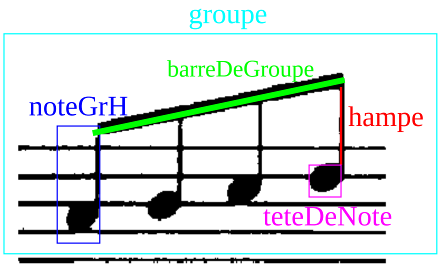

FIG. 1 - *Représentation graphique de la règle groupe*

Les variables de grammaire suivantes représentent :

fracSoupir	demi-soupir, quart de soupir...
noteGr	note graphique
noteGrH	note graphique à hampe orientée vers le haut
noteGrB	note graphique à hampe orientée vers le bas
têteDeNote	tête de note avec tous ses attributs
tête	pouvant être une tête de noire ou de blanche
altération	pouvant être bécarre, bémol ou dièse

Les connecteurs de positionnement utilisés dans cet exemple ont les significations suivantes :

extrémitéGauche	à l'extrémité gauche
procheV	proche selon la verticale
extrémitéDroite	à l'extrémité droite
procheGExtBas	proche à gauche de l'extrémité basse
procheGmêmeLigne	proche à gauche sur à la même hauteur

Les règles sont :

groupe	::= barreDeGroupe (**extrémitéGauche** noteGr :
	[**procheV** [noteGr | fracSoupir]]* :
	extrémitéDroite noteGr)
noteGr	::= noteGrH | noteGrB
noteGrH	::= hampe **procheGExtBas** têteDeNote
têteDeNote	::= tête **procheGmêmeLigne** [altération]?
hampe	::= *segment*
barreDeGroupe	::= *segment*

Les règles, barreDeGroupe et hampe provoquent un étiquetage des segments leur correspondant. On peut donc ainsi traduire la règle NoteGrH par :

> On a une noteGrH si : on a une hampe, et proche à gauche de l'extrémité basse de la hampe, on trouve une têteDeNote.

L'intérêt de la notation de factorisation est illustrée par la règle Groupe. Cette règle signifie qu'un groupe est constitué d'une barre de groupe et qu'à chaque extrémité de cette barre il y a une note graphique et qu'entre ces deux notes il peut y avoir d'autres notes graphiques. Ainsi, chaque connecteur de positionnement (**extrémitéGauche**, **procheV** et **extrémitéDroite**) fait référence au terminal BarreDeGroupe. Ceci est utile lorsque plusieurs objets sont positionnés par rapport à un seul autre.

Voyons maintenant la partie graphique de la grammaire de manière plus globale à l'aide de la figure 2. Nous pouvons constater qu'à ce niveau, il n'y a pas de prise en compte de la polyphonie. Les notes sont simplement reconnues de manière indépendante et ce n'est que dans la partie syntaxique que ces notes seront réparties voix par voix.

Dans cette partie graphique, seuls les attributs des notes sont reconnus avec les notes, c'est-à-dire : les altérations, les signes d'attaques et les points d'allongement de durée. Les autres signes, comme les nuances, qui se rapportent à une voix ou à la partie entière seront reconnus par la suite dans la partie syntaxique.

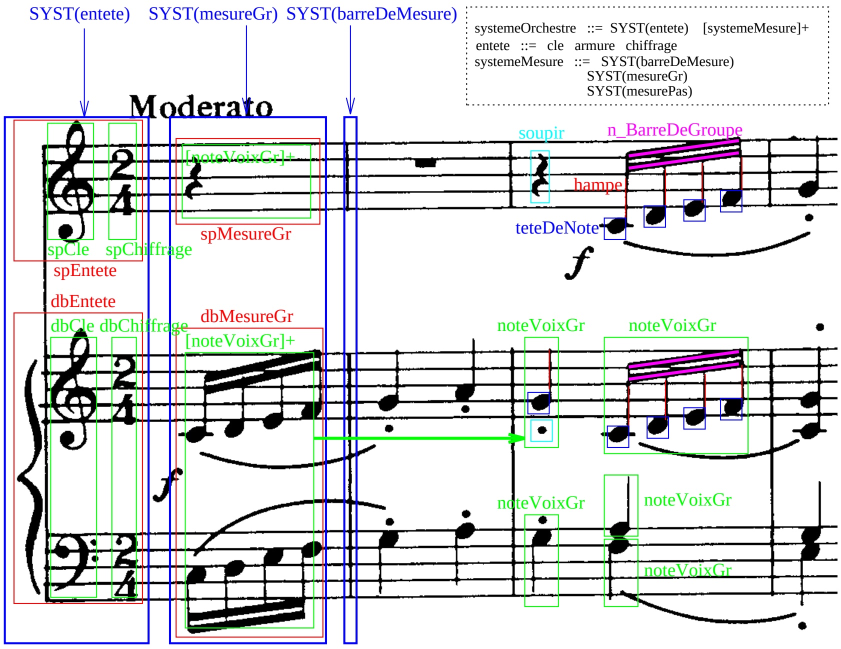

FIG. 2 - *Représentation graphique du niveau partition et de mesureGr*

6.4 Présentation de la partie syntaxique

Cette partie utilise les résultats de la partie graphique, c'est-à-dire l'ensemble des notes reconnues et de leurs attributs. Par rapport à ces notes, sont définis les positionnements des attributs de voix ou de portée comme les liaisons de phrasées, les clés en cours de mesure, (et donc éventuellement au milieu d'un groupe), les nuances, les alta et bassa... Les notes sont en plus représentées voix par voix ce qui permet de réaliser la vérification de la cohérence entre le nombre de temps dans la mesure et le chiffrage.

Afin de réaliser une vérification de l'alignement vertical des notes, aussi bien entre les portées d'une partition d'orchestre, mais aussi entre les voix d'une portée polyphonique, nous proposons la notion de Pas 3. Un Pas correspond à un découpage temporel de la partition, ce qui graphiquement donne un découpage vertical de la partition. La durée d'un Pas est en fait la plus petite durée dans une colonne de notes verticalement alignées (c'est-à-dire débutant en même temps) sur l'ensemble de la partition. Nous verrons par la suite (section 7) plus précisément comment ces erreurs sont détectées.

6.5 Présentation du niveau partition

Nous avons vu dans la description de la structure d'une partition (section 6.1) qu'il était nécessaire de considérer les portées de type piano, non pas comme deux portées indépendantes, mais comme une seule portée à 11 lignes. Ceci afin de permettre à une voix de passer d'une portée à l'autre. Les règles de grammaire s'attachant à une portée seront donc toujours décomposées en une règle pour les portées simples et une pour les portées dites doubles, comme dans cet exemple :

$$\text{mesureGr} ::= \text{spMesureGr} \mid \text{dbMesureGr}$$

Lorsqu'il y a notion de voix dans une règle, les portées simples sont limités à 3 voix et les portées doubles à 5 voix.

Nous pouvons voir sur les figures 2 et 3 également l'utilisation de l'opérateur SYST afin de passer automatiquement d'une règle pour une portée seule (simple ou double) à une règle pour l'ensemble des portées d'un système d'orchestre.

Les difficultés de segmentation liées aux objectifs et les solutions pour les résoudre grâce à cette grammaire, sont présentées dans [6].

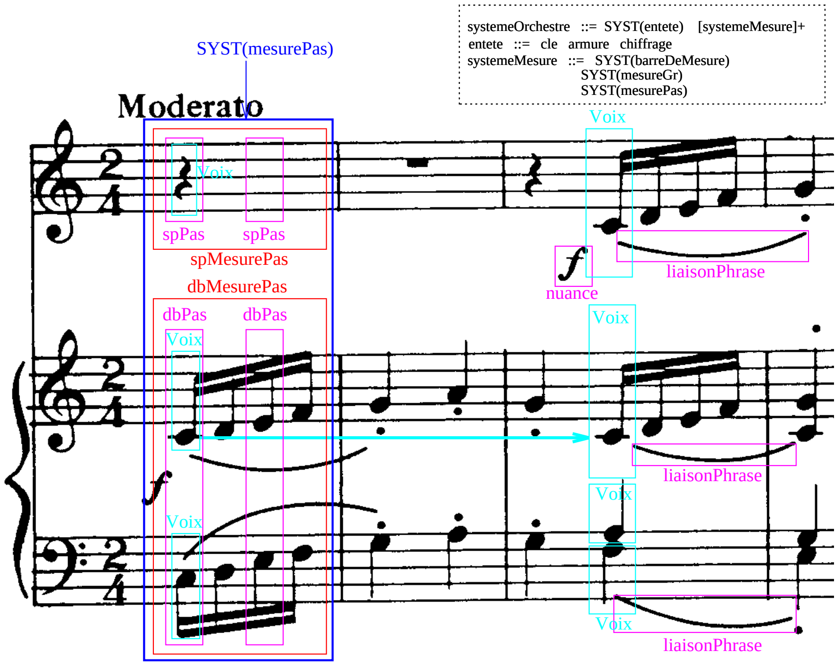

FIG. 3 - *Représentation graphique de mesurePas*

7 Utilisation de la grammaire pour la détection d'erreurs temporelles

Pour obtenir un système de reconnaissance de partition utilisable et efficace, il est nécessaire que celui-ci soit fiable. Fiable dans le sens où ce qui est produit par le système ne nécessite pas de contrôle ultérieur. Ceci est possible car l'information musicale est redondante, ce qui permet différents contrôles automatiques de ce qui est reconnu. Cependant, cette information musicale doit être utilisée au maximum pour être redondante. Or, c'est effectivement le cas dans la grammaire que nous avons présentée. Les erreurs à détecter peuvent être de deux ordres :

- celles liées au système de reconnaissance,
- celles liées à la qualité du document analysé. En effet, les documents scannés ne sont pas toujours parfaits, il y a parfois des parties de l'image abimées, ou mal imprimées à l'origine, faisant disparaître par exemple une hampe ou transformant une tête de blanche en noire. D'autres défauts d'impression font apparaître des zones «poivre et sel» dans l'image, rendant impossible la reconnaissance d'un objet. Un musicien compense ce manque de qualité par le contexte musical. Ce type d'objet qui pris séparément est non reconnaissable, le devient en utilisant le contexte musical.

Mis à part le contexte graphique tel qu'il est défini dans la grammaire, le contexte musical peut être syntaxique ou sémantique.

Lorsqu'il est syntaxique, on peut utiliser sans difficulté le nombre de temps dans une voix d'une mesure comme devant être égal au chiffrage. Et on peut également utiliser la forte information comme quoi les notes débutant en même temps sont verticalement alignées dans une partition d'orchestre. C'est dans la partie syntaxique de la grammaire que ces informations seront utilisées.

Lorsqu'elle est sémantique, l'information est plus sujette à variation suivant le compositeur. En effet, ces informations sont liées aux règles d'harmonie, ou par exemple au fait que s'il y a un motif rythmique pour un instrument, il y a de fortes chances de retrouver ce motif sur les autres instruments au même moment. Mais, ces règles sont trop restrictives car un compositeur souhaitera par exemple avoir des rythmes concurrents et différents pour obtenir un effet particulier.

Nous nous limiterons au contexte syntaxique qui est suffisamment généralisable (même si très rarement certains compositeurs ne suivent pas par exemple la règle du nombre de temps de la mesure) et de plus permet de détecter les mêmes erreurs qu'en utilisant les motifs sans le désavantage du manque de liberté laissé au compositeur.

La détection des erreurs du nombre de temps dans une voix par mesure se fait simplement en calculant la somme des durées dans une voix et en la comparant au chiffrage courant. Ceci se fait à l'aide d'attributs ajoutés à la grammaire dans la partie syntaxique. Ces attributs «horizontaux» circulant correctement grâce à l'opérateur SYST.

La détection des incohérences alignement-durée des notes est plus compliquée. En effet, il faut comparer la position d'une note avec toutes celles qui lui seront alignées sur les autres voix ou les autres portées, et vérifier que le temps déjà écoulé dans une voix jusqu'à cette note est cohérent avec les autres voix ou portées.

Pour ce faire, nous utilisons la notion de Pas de la grammaire. Afin de pouvoir comparer la position (temporelle/graphique) d'une note avec celle des autres, nous proposons une grille qui contient pour chaque Pas, la position des notes alignées et la durée du Pas, c'est-à-dire la durée de la note la plus courte débutant sur ce Pas. De ce fait un Pas contient au moins une note débutant à ce pas et éventuellement des notes se poursuivant, car ayant débuté à un Pas précédent en étant de durée plus longue que ce dernier. Nous obtenons ainsi une structure représentant un découpage temporel de la partition, où à chaque Pas est associé sa durée idéale compte-tenu de toutes les autres notes du Pas.

Cette grille est calculée lorsque toutes les notes de la mesure de la partition d'orchestre ont été reconnues, c'est-à-dire entre SYST(mesureGr) et SYST(mesurePas). Cette grille possède suffisamment d'informations pour qu'en comparant simplement la durée de la grille avec la durée de la note dans le Pas, on puisse détecter des incohérences alignement-temps sans avoir à comparer la durée de la note avec celle de toutes les autres du même Pas, permettant ainsi de ne vérifier manuellement que les parties détectées par le système, et non pas l'ensemble de la portée.

Il est de plus possible de proposer une correction de la durée d'une note lorsqu'une erreur est détectée, en proposant la durée calculée par l'intermédiaire de la grille.

Dans le futur, il serait possible de retourner dans l'image pour vérifier la présence d'une note non reconnaissable mais détectée par l'intermédaire de la grille, et ainsi améliorer les capacités de reconnaissance du système.

8 Résultats

La grammaire est actuellement entièrement définie pour des partitions d'orchestre, avec différentes voix sur une même portée, des accords dans une voix, des signes d'attaque, n-olets (triolet...), pause, octava, nuances, liaisons rythmiques et de phrasé, et appogiatures. Les abréviations, ornements (sauf les appogiatures) et les paroles ne sont pas encore inclus dans la grammaire.

La plupart des outils nécessaires pour implémenter la grammaire ainsi que les opérateurs de position ont été écrits. Nous avons développé un compilateur de grammaire permettant, partant de la grammaire, de produire automatiquement l'analyseur associé. Grâce à cette séparation de la connaissance a priori et du programme, il est possible de tester le système sur un sous-ensemble de la grammaire sans avoir de problèmes lors du passage à la grammaire complète.

Nous avons donc développé le système en utilisant un sous-ensemble de grammaire permettant de reconnaître les partitions d'orchestre polyphoniques avec les clés, chiffrages, armures, blanches, noires, croches simples, groupes de notes, altérations, points d'allongement de durée, pauses et barres de mesure. Ce système peut, en utilisant le contexte, résoudre des problèmes de segmentation lorsqu'un symbole est connexe à un groupe de notes. Il permet également de détecter et éventuellement corriger, la plupart des erreurs temporelles grâce à l'alignement vertical et le nombre de temps dans une voix par mesure.

En exemple, pour analyser une image de 1900x900 pixels, comportant une partition d'orchestre à 3 portées possédant 78 éléments reconnus (sur 282 éléments analysés), sur un SUN Sparc 10, il faut 6s pour le traitement d'image (extraction des segments, détection et suppression des lignes de portée, construction de la structure à analyser) et 5s pour l'analyse au niveau grammatical.

9 Conclusion

Nous avons présenté un système de reconnaissance de partitions musicales entièrement controlé par une grammaire. Cette grammaire a été définie pour pouvoir reconnaître des partitions d'orchestre, avec différentes voix sur une seule portée, des accords, des signes d'attaque, des liaisons, des nuances...

Contrairement aux autres systèmes, notre approche formalise toutes les règles utilisées pour la reconnaissance et produit une intégration maximum du contexte. Ce contexte permet d'améliorer la reconnaissance en réglant les de difficiles problèmes de segmentation (objets touchant d'autres objets) [6]. Habituellement, la syntaxe est utilisée pour vérifier un étiquetage. Ici, c'est la syntaxe qui controle tout le processus de reconnaissance produisant ainsi un meilleur étiquetage.

Un autre avantage de cette formalisation grammaticale, est la séparation entre la partie dynamique du système (le programme) et la définition des règles musicales (la grammaire). Cette séparation nous permet de modifier facilement une règle ou d'adapter le système à un autre type de document structuré. En effet, il suffit de définir une nouvelle grammaire pour construire un nouveau système de reconnaissance.

De plus, cette méthode est une réponse possible aux problèmes non résolus décrits par Blostein [2] et Carter [4] : reconstruction d'objets séparés et segmentation d'objets se touchant. Elle permet également de résoudre les difficultés de passage d'un prototype à un système complet. Le système est actuellement capable de reconnaître des partitions d'orchestre polyphoniques. Grâce au nombre de temps dans une mesure et à l'alignement vertical des notes, l'analyseur peut détecter et corriger des erreurs sur la durée des notes, afin de produire des résultats plus fiables.

Références

[1] A. Andronico and A. Ciampa. On automatic pattern recognition and acquisition of printed music. In *ICMC, International Conference on Computer Music*, pages 245–278, Venice, Italy, 1982.

[2] Dorothea Blostein and Henry Baird. A critical survey of music image analysis. In Springer-Verlag, editor, *Structured Document Image Analysis*, pages 405–434, Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[3] Nicholas P. Carter. A new edition of walton's façade using automatic score recognition. In H. Bunke, editor, *Advances in Structural and Syntactic Pattern Recognition (Proceedings of International Workshop on Structural and Syntactic Pattern Recognition, Bern, CH)*, pages 352–362, World Scientific, 1992.

[4] Nicholas P. Carter and Richard A. Bacon. Automatic recognition of printed music. In Springer-Verlag, editor, *Structured Document Image Analysis*, pages 456–465, Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[5] N.P. Carter, R.A. Bacon, and T. Messenger. The acquisition representation and reconstruction of printed music by computer : a review. *Computers and the Humanities*, 22:117–136, 1988.

[6] Bertrand Couasnon and Jean Camillerapp. Using grammars to segment and recognize music scores. In *International Association for Pattern Recognition Workshop on Document Analysis Systems*, pages 15–27, Kaiserslautern, Germany, October 1994.

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

[8] Ichiro Fujinaga, Bo Alphonce, Bruce Pennycook, and Natalie Boisvert. Optical recognition of music notation by computer. *Computers in Music Research 1*, 161–164, 1989.

[9] Hirokazu Kato and Seiji Inokuchi. A recognition system for printed piano music using musical knowledge and constraints. In Springer-Verlag, editor, *Structured Document Image Analysis*, pages 435–455, Eds. H.S. Baird, H. Bunke, K. Yamamoto, 1992.

[10] Philippe Martin. *Réseaux de neurones artificiels: Application a la reconnaissance optique de partitions musicales.* PhD thesis, IMAG, Grenoble, France, 1992.

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

[12] J. W. Roach and J.E. Tatem. Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment. *Pattern Recognition*, 21(1):33–44, 1988.