

Reconnaissance automatique d'accords et modélisation des logiques d'enchaînement dans un logiciel d'improvisation

Rapport de stage
José Echeveste

Avril - Juillet 2010
EHESS - IRCAM
Maître de stage : Marc Chemillier

Remerciements

Je voudrais tout d'abord exprimer ma sincère reconnaissance à Marc Chemillier pour la confiance qu'il m'a accordée dès notre première rencontre, son aide et son soutien tout au long du stage.

Mon travail s'est déroulé en partie à l'Ircam et je tiens à remercier l'institut pour son accueil, les membres de l'équipe « Représentations musicales » pour leur sympathie et leurs idées constructives notamment Gérard Assayag, Jean Bresson et Benjamin Levy.

Un grand merci également à Sylvie Benoit qui m'a chaleureusement accueilli dans son bureau.

Résumé

Ce rapport présente un programme de reconnaissance harmonique automatique à partir de notes midi intégrées au logiciel d'improvisation OMax ainsi qu'un module utilisant un corpus de partitions du musicien/compositeur Hermeto Pascoal permettant entre autres choses la génération d'enchaînements d'accords et l'harmonisation de mélodies dans le style du musicien. Le programme de reconnaissance est inspiré de l'algorithme de Pardo et Birmingham qui se différencie des autres algorithmes de reconnaissance par sa simplicité. Le programme est adapté à un contexte d'improvisation jazz et à un fonctionnement « avec Beat » (les notes midi situées entre deux mêmes pulsations sont regroupées) pré-existant du logiciel OMax. Plusieurs éléments de ce programme de reconnaissance sont ensuite repris pour l'harmonisation automatique dans le style d'Hermeto Pascoal.

Mots-clés

OMax, OpenMusic, reconnaissance d'accords, analyse harmonique, harmonisation, Musique Assistée par Ordinateur.

Table des matières

Remerciements	1
Résumé	2
Introduction	6
1 Contexte	7
1.1 Etat de l'art	7
1.2 Omax	8
2 Réalisation	12
2.1 Reconnaissance automatique d'accord	12
2.1.1 Algorithme de Pardo et Birmingham	12
2.1.2 Adaptation de l'algorithme et réalisation	13
2.2 Caler un extrait d'une improvisation sur une grille connue . .	17
2.3 Utilisation de l'harmonie d'Hermeto Pascoal	18
2.3.1 Fonctions	18
2.3.2 Harmonisation dans le style d'Hermeto Pascoal	19
2.3.3 Harmonisation dans le style d'Hermeto Pascoal avec indication sur l'accord précédent	19
3 Futures directions	21
Conclusion	22

Table des figures

1.1	Architecture du système OMax	10
1.2	Exemple d'objet «beat»	11
2.1	Segmentation	14

Liste des algorithmes

1	Algorithme de calcul du poids d'un accord par rapport à un beat	15
---	--	----

Introduction

Ce stage s'insère dans le programme de recherche sur les nouvelles technologies appliquées à l'improvisation défini par le projet ANR 2009 « IMPROTECH ». Il vise à expérimenter certaines modalités d'interaction avec des musiciens improvisateurs en développant un prototype logiciel qui intègre des connaissances musicales harmoniques.

Le logiciel d'improvisation OMax développé à l'IRCAM capte le jeu d'un musicien en direct et produit une improvisation à partir des phrases jouées par celui-ci, seul ou en interaction. La conception d'OMax est purement « agnostique », c'est-à-dire que le logiciel ne comporte aucune connaissances musicales préétablies.

Parmi les prototypes ayant précédé l'apparition d'OMax, certains intégraient des connaissances musicales particulières permettant notamment à l'ordinateur de se caler sur une pulsation régulière et de suivre des enchaînements harmoniques. Le but du projet est de renouer avec cette approche en intégrant dans un programme de simulation de l'improvisation, des fonctions de reconnaissance d'accords, de génération d'enchaînement harmonique et d'harmonisation automatique.

Après un bref aperçu des différentes techniques existantes autour de la reconnaissance automatique d'accords nous étudierons en détail la réalisation et les résultats de ce projet. Nous proposerons également quelques directions futures envisageables .

Chapitre 1

Contexte

1.1 Etat de l’art

L’analyse harmonique automatique a fait l’objet de nombreux travaux de recherche abordés sous différentes approches : modèles probabilistes, systèmes de règles ou correspondances de motifs (model-matching), dans un but pédagogique, d’aide à la composition...

Dans le domaine de la reconnaissance d’accords depuis le signal audio, de nombreux travaux utilisent des modèles de Markov cachés [5, 3]. Dans ces travaux, une phase d’apprentissage est nécessaire, afin d’entraîner le système sur une base de données annotée. Différents paramètres comme la probabilité de transition d’un accord à un autre, ou la probabilité d’un accord suivant un descripteur (tel les chromas, qui représentent l’intensité des fréquences correspondant aux 12 demi-tons ramenés sur une octave) sont définis lors de cette phase. Ils sont ensuite utilisés lors de la phase de reconnaissance, qui utilise la programmation dynamique pour trouver le meilleur chemin dans le graphe des accords candidats. Les chromas peuvent également être utilisés à la manière d’un profil de notes, en cherchant ensuite à maximiser une corrélation avec un vecteur de référence correspondant à un accord, comme dans les travaux de Gomez [2].

Dans le domaine de la musique symbolique, on citera le « modèle en spirale » de Chew [1], qui en plus de la tonalité, peut estimer la suite d’accords d’un morceau en utilisant une fenêtre d’analyse. D’autres travaux utilisent un système de règles pour déterminer les accords d’après les notes de musique, comme Melisma Music Analyzer, le logiciel proposé par Temperley [11], et

l'approche récemment proposée par Illescas et al. [4]. Celle-ci est intéressante et pourrait servir d'appui pour le sage même si elle n'est pas adaptée à une application « temps réel ». Elle se divise en 5 étapes distinctes, efficaces et complémentaires les unes avec les autres. Détaillons ici cette approche.

Elle effectue d'abord une analyse mélodique afin d'identifier les notes comme faisant partie ou non de l'harmonie. Un ensemble de règles est défini pour permettre la caractérisation des notes de passages, des appoggiatures, des retards... avec un certain degré de confiance. On obtient alors une liste de différentes analyses mélodiques possibles. Dans un deuxième temps, chaque mesure est divisée en fenêtres temporelles dont la durée est égale à la plus courte valeur rythmique (exemple : la double-croche s'il y a dans la mesure des doubles-croches, des noires et une blanche). On construit pour chacune de ces fenêtres, à partir des notes qu'elles contiennent, un ensemble d'accords possibles. Pour chaque fenêtre un ensemble de règles permettent l'élimination des tonalités ne pouvant correspondre aux notes présentes, parmi les 24 possibles. Celles qui restent permettent la construction d'un graphe pondéré orienté acyclique. Chaque niveau du graphe correspond à une fenêtre ; un noeud correspond à un accord avec sa fonction tonale dans une certaine tonalité ; les arcs relient les noeuds appartenant à des couches successives de façon à former des progressions valides. Les arcs sont pondérés en fonction des cadences qu'elles induisent. La dernière étape consiste à calculer grâce à une approche de programmation dynamique, le chemin de plus faible poids, correspondant à l'analyse harmonique la plus plausible.

Rocher et al. [9] proposent une méthode d'estimation d'accord reposant sur la comparaison d'un profil obtenue à partir de la fréquence d'apparition des notes sur une période donnée, de leur durée et de leur contribution dans l'harmonie, avec un profil de référence.

Pardo et Birmingham [7] ont développé un système qui partitionne la musique tonale en segments harmoniques correspondant à une même harmonie. Ces segments sont ensuite étiquetés avec les labels d'accords appropriés. Le système est d'une efficacité surprenante malgré sa simplicité.

1.2 Omax

Le système OMax génère des improvisations en temps-réel. Il utilise d'une part, une modélisation par grammaire formelle des règles de substitution d'accords dans le Jazz et d'autre part une structure formelle d'apprentissage,

l'oracle des facteurs. Ce système est entre deux paradigmes : temps-réel avec l'utilisation du logiciel Max/Msp pour recevoir les notes d'un clavier midi et jouer les données, et pseudo temps-réel avec les calculs musicaux symboliques gérés par le logiciel OpenMusic, basé sur le langage Common Lisp Object System (Figure 1.2).

Il existait dans le système OMax un mode «Beat» qui segmentait la pulsation et imposait le contexte harmonique. Ce mode générait un accompagnement à l'improvisateur : le tempo était fixe, la basse, la batterie et le clavier générés automatiquement jouaient un accompagnement autour de la grille d'accords de référence du morceau. Ce mode utilisait une classe objet «beat». Un «beat» correspond à un ensemble de notes midi contenues dans un temps rythmique avec leur harmonie associée (Figure 1.2).

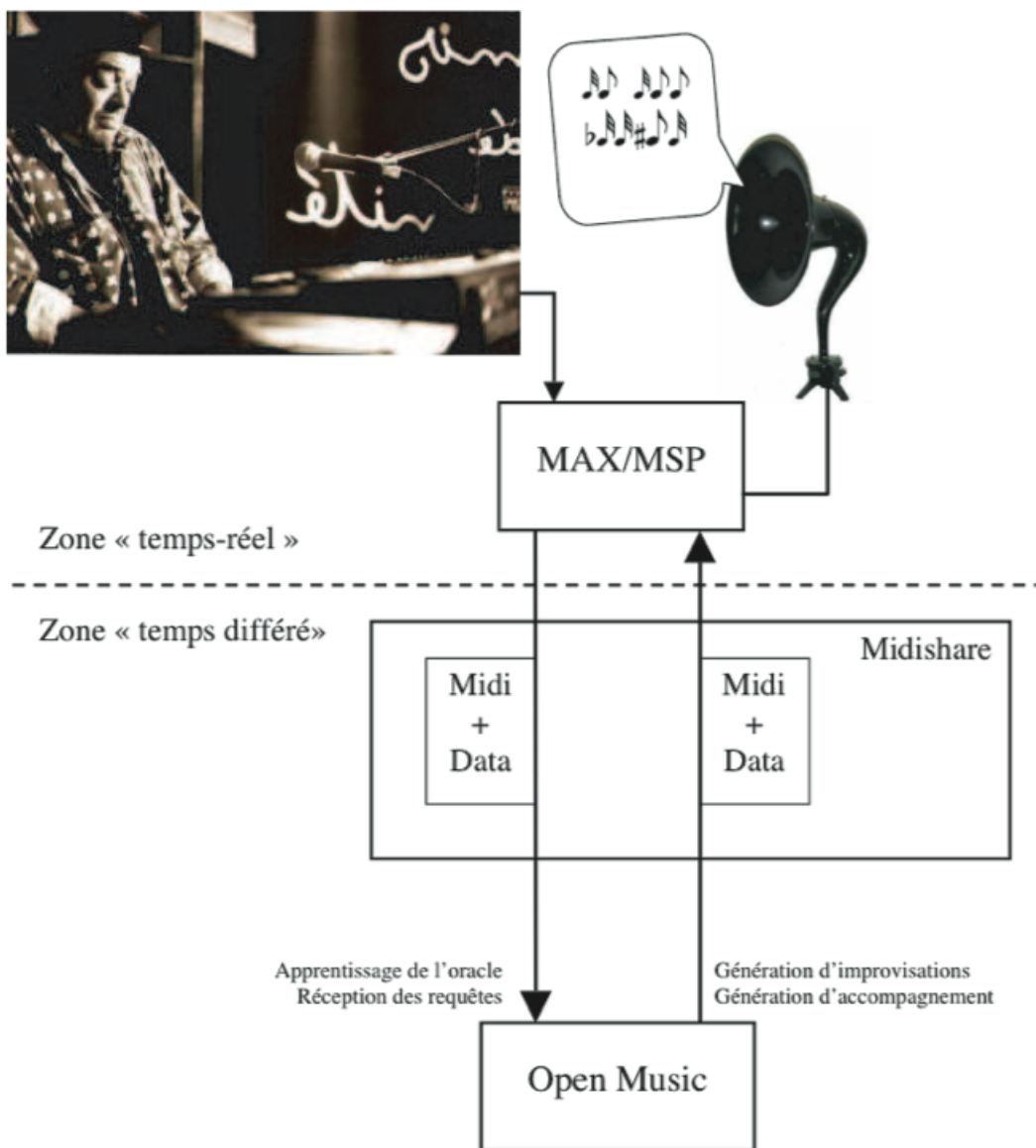


FIGURE 1.1 – Architecture du système OMax

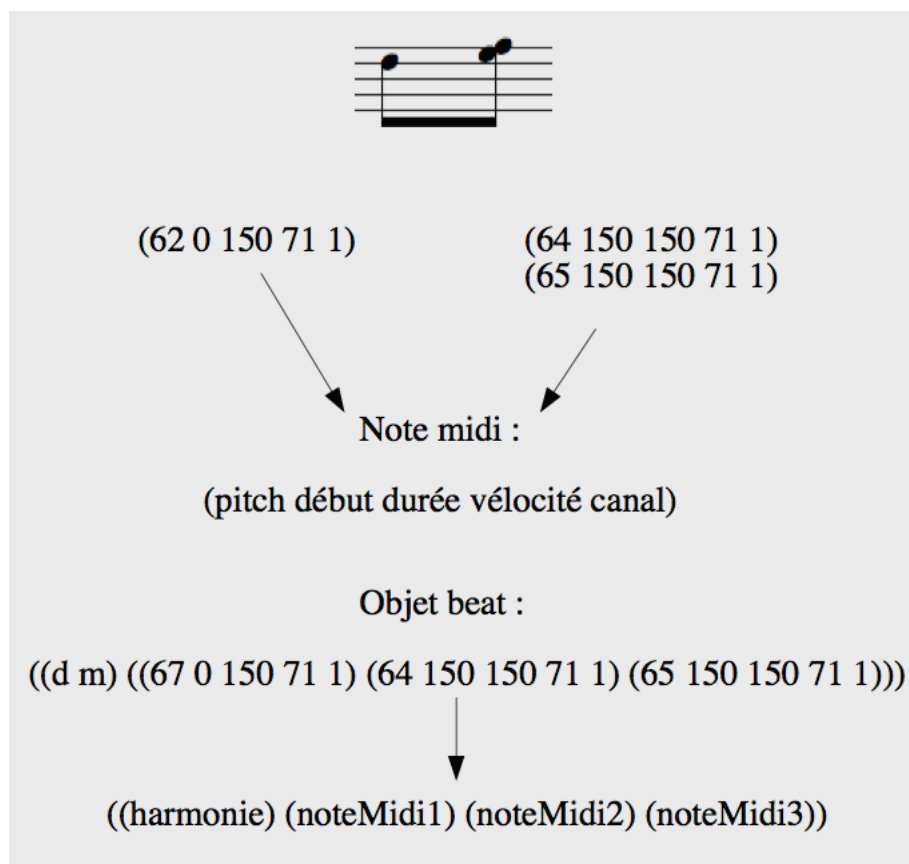


FIGURE 1.2 – Exemple d'objet «beat»

Chapitre 2

Réalisation

2.1 Reconnaissance automatique d'accord

Dans les précédents travaux, les accords dans les objets «beat» étaient insérés «manuellement». Le but de cette partie du projet est d'insérer dans ces objets les accords de façon automatique. Cela donne la possibilité de générer un accompagnement qui s'adapte à l'harmonie de l'improvisateur et non le contraire. La réalisation de ce travail est basé sur les travaux de Pardo et Birmingham [7], leur algorithme nous a semblé le plus approprié et le plus facilement adaptable à un contexte improvisation jazz / traitement « temps réel ».

2.1.1 Algorithme de Pardo et Birmingham

Cette algorithme sépare le problème de la reconnaissance automatique d'accords en deux parties : la segmentation et l'étiquetage proprement dit. Il est adaptable à un contexte « temps réel » car la segmentation et l'étiquetage peuvent être établis définitivement, indépendamment du futur.

L'algorithme qui permet l'étiquetage des accord est très simple. Chaque note correspond à un nombre : Do 0, Do# 1, Ré 2,..., Sib 10, Si 11. Les accords sont modélisés sous la forme d'une liste de ces nombres. Par exemple, la triade de do majeur composée de la tonique (do), d'une tierce majeure (mi) et d'une quinte juste (sol) sera représenté sous la forme (0 4 7). Les différents types d'accords (mineur, majeur, 7ème de dominante...) sont définis sous cette forme en prenant do comme tonique. Les accords dans les autres tonalités sont obtenus en additionnant modulo 12 chaque membre de la liste par le numéro

correspondant à la tonique du nouvel accord. Par exemple pour obtenir la triade de fa majeur :

$$(0+5 \ 4+5 \ 7+5) = (5 \ 9 \ 0) \text{ donc dans l'ordre } (0 \ 5 \ 9)$$

Pour trouver l'harmonie d'un segment (ensemble des notes présentes entre deux instants donnés), on calcule à partir des hauteurs des notes du segment, pour tous les types d'accords, avec toutes les toniques possibles, un poids. Il est égal à la somme des poids de chaque note présente dans l'accord (qui peut dépendre de leur durée), moins la somme du nombre de notes qui ne sont pas dans l'accord, moins la somme du nombre de notes de l'accord qui ne sont pas dans les notes du segment. On associe l'harmonie du segment à l'accord ayant le poids le plus fort.

La segmentation consiste quant à elle à délimiter les durées des différentes harmonies. Deux algorithmes sont proposés par les auteurs de l'article de complexité quadratique pour l'un, linéaire pour l'autre avec des résultats presque similaires. L'idée de l'algorithme de moindre complexité est la suivante :

On parcourt un à un les segments minimaux ; si le poids maximum du segment que l'on traite + le poids maximum du segment précédent est inférieur au poids maximum des deux segments réunis alors on fusionne les 2 segments.

La figure 2.1.1 montre un exemple de segmentation. À chaque étape on ne considère qu'un seul sommet (point de partition). Les flèches noires épaisses représentent les segments dont le poids a été calculé à cette étape. Les flèches grises représentent les segments dont le poids a été calculé à une étape précédente. Le nombre dans un noeud correspond à la somme des poids du meilleur chemin, du début jusqu'à ce noeud.

En combinant ces 2 techniques, segmentation et pondération, Pardo et Birmingham ont conçu un algorithme simple et efficace, facilement adaptable à un contexte particulier.

2.1.2 Adaptation de l'algorithme et réalisation

L'ensemble des modifications de l'algorithme de Pardo et Birmingham pour notre projet ont été réalisé pour deux raisons :

- le fait que l'algorithme soit à priori destiné à une analyse plus classique que jazz de la musique
- l'utilisation des objets «beat» préexistants dans OMax.

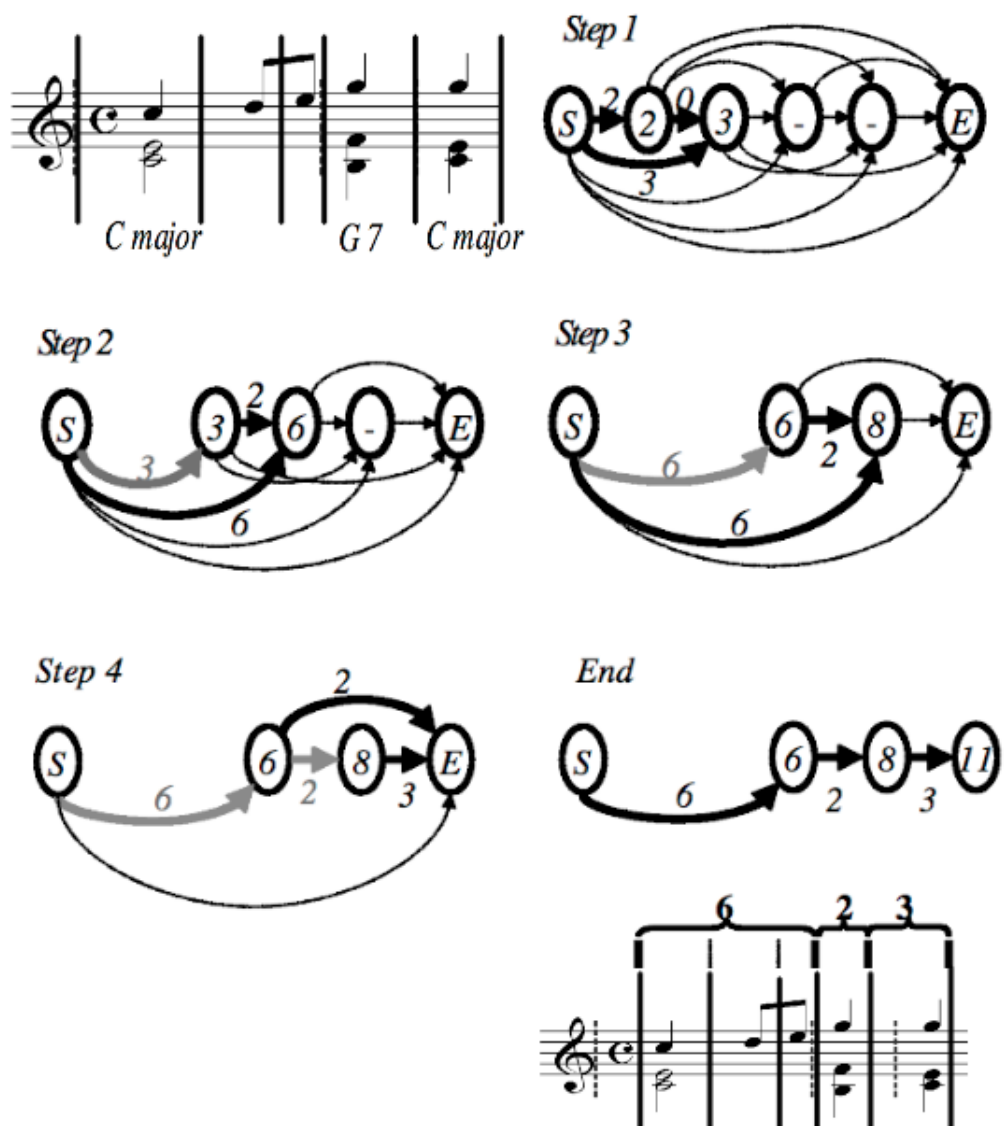


FIGURE 2.1 – Segmentation

Segmentation

Notre souhait est de pouvoir effectuer une analyse harmonique similaire aux grilles de standards du jazz. Il nous semble judicieux de définir le segment minimal comme la durée d'une pulsation, c'est-à-dire la durée d'un « beat ». En effet la fonction de reconnaissance automatique prend en entrée une improvisation midi sous la forme d'une liste de quintuplet (pitch début durée vélocité canal) et la pulsation associée ; le résultat correspond à une liste d'objets «beat», c'est-à-dire une liste de quintuplets entre deux pulsations successives avec leur label harmonique associé.

Ajout d'accords harmoniquement plus riches

La liste des différents types d'accords proposée par Pardo et Birmingham est assez réduite : accord parfait majeur, accord parfait mineur, accord de septième de dominante, accord de quinte diminuée, accord de septième diminuée, accord demi-diminué. Or dans le jazz d'autres accords apparaissent aussi souvent que ceux de cette liste. On décide donc d'ajouter les accords suivants : accord de septième majeur, accord mineur sept, accord sept bémol 9, accord altéré. Une plus grande liste est difficile à mettre en place car les accords candidats sont équivalents aux accords déjà présents du point de vue des intervalles à un renversement près. D'autre part on sait que dans le jazz les extensions des accords sont utilisées de manière quasi systématique. C'est pourquoi on étend la caractérisation de certains accords avec les 9ème et 13ème dans leur définition.

Prise en compte de la durée des notes

L'algorithme de Pardo et Birmingham ne tient pas compte de la durée des notes mais seulement de leur présence dans les segments minimaux. Dans notre implémentation, la durée des notes est prise en compte au niveau de la fonction de calcul du poids. Cette fonction prend en entrée les couples pitch-durée présents dans le «beat» et l'accord dont on veut calculer le poids. Le poids est calculé de la façon suivante :

somme des durées de chaque note présente dans l'accord

- somme des durées des notes qui ne sont pas dans l'accord

- somme du nombre de notes de l'accord qui ne sont pas dans les notes du segment * la durée minimale des notes présentes dans le «beat». D'autres variantes ont été testées. Par exemple en multipliant la somme du nombre

de notes de l'accord qui ne sont pas dans les notes du segment par la durée maximale, la durée moyenne des notes présentes dans le «beat». Les résultats étaient équivalents ou meilleurs pour la solution avec la durée minimale.

Algorithme 1 Algorithme de calcul du poids d'un accord par rapport à un beat

Entrées: *accord* : liste de pitch, *beat* : liste de couples (pitch,durée)

incount $\leftarrow$ 0
outcount $\leftarrow$ 0
pour tout *n* dans *beat* **faire**
 si pitch de *n* $\in$ *accord*
 incount += durée de *n*
 sinon
 outcount += durée de *n*
 finsi
fin boucle
missingcount $\leftarrow$ nombre de notes de l'accord non présentes dans le beat
 * min des durées
Retourner *incount* $-$ (*outcount* + *missingcount*).

Gestion en cas d'égalité

Une fonction permet de favoriser certains enchaînements harmoniques en cas d'égalité dans les poids des accords pour un segment. En effet on mémorise la fondamentale du segment précédent pour favoriser les enchaînements d'accords de même fondamentale puis les enchaînements d'accords dont les basses forme un intervalle de quinte (Ier degré vers Vème degré , IIème vers Vème ...).

Résultats

En effectuant les tests sur cinq grilles du solo de Bill Evans sur Israel (main droite + main gauche) on obtient environ 25% d'accords identiques à ceux de la grille originale. On remarque cependant qu'il n'y a jamais de résultat absurde au niveau harmonique. En effet soit l'accord trouvé est dans le même mode que l'accord attendu soit l'erreur est due à un retard ou une anticipation volontaire du musicien par rapport à la grille.

2.2 Caler un extrait d'une improvisation sur une grille connue

En général lors d'un enregistrement ou d'un concert de musique jazz, on connaît la structure harmonique du morceau que l'on va jouer. La reconnaissance harmonique d'un extrait d'une improvisation devient alors plus simple. On a donc implémenté une fonction qui prend en entrée une grille d'un morceau sous forme de beat harmonique et un extrait de solo et qui doit identifier la section de la grille harmonique qui correspond avec l'extrait de l'improvisation. Expliquons brièvement le principe :

On utilise la fonction de calcul du poids de l'accord par rapport à un «beat» vu précédemment.

On commence par calculer le poids de l'accord du premier beat harmonique de la grille avec le premier beat mélodique du solo, du deuxième beat harmonique avec le deuxième beat mélodique... On additionne ces poids pour obtenir un poids global traduisant la probabilité pour que le début du solo corresponde au début de la grille harmonique.

On calcule ensuite le poids de l'accord du second beat harmonique de la grille avec le premier beat mélodique du solo, du troisième beat harmonique avec le deuxième beat mélodique... On additionne ces poids pour obtenir un poids global traduisant la probabilité pour que le début du solo commence au deuxième accord de la grille.

On essaye toutes les correspondances possibles entre beats harmoniques successifs et beats mélodiques successifs. L'enchaînement d'accord ayant le poids le plus élevé est l'élus.

Résultats

On obtient des résultats proches de 100% pour le solo de Bill Evans. En revanche lorsque le solo en entrée est sans accompagnement main gauche (par exemple sur un solo du saxophoniste Art Pepper), les résultats ne sont pas aussi convaincants.

2.3 Utilisation de l’harmonie d’Hermeto Pascoal

Hermeto Pascoal (né le 22 juin 1936) est un compositeur et multi-instrumentiste brésilien. Les musicologues s’accordent à dire qu’il possède un jeu harmonique complexe qui lui est propre. Une analyse musicale de plusieurs de ses partitions, les consultations de notes de musiciens ayant joué avec lui, et d’un rapport de thèse le concernant [6], différents entretiens avec Jean-Pierre Cholleton (Professeur au conservatoire de Montreuil) nous ont permis de mettre en évidence certaines de ses caractéristiques harmoniques (utilisation quasi systématique d’accords mineur 4 7 9 en guise de second degré, enchaînement d’accords parallèles, basse changeante sur un même accord...). Nous souhaitons intégrer le style harmonique du compositeur au logiciel OMax d’une certaine façon. Des fonctions de substitution d’accords conformes à la grammaire de Steedman [10] étaient déjà intégrées au logiciel. Elles correspondaient à des règles de substitution d’accords du jazz BeBop.

L’idée de départ était de repérer des règles similaires caractérisant le jeu harmonique d’Hermeto Pascoal grâce à une analyse approfondie de sa musique, plus particulièrement en comparant l’harmonie de standards du jazz avec les réharmonisations de ces mêmes morceaux. Mais les quelques régularités trouvées ne nous ont pas permis d’en tirer des règles générales de substitution. Nous avons donc décidé d’importer dans l’environnement Open Music un grand nombre de grilles issues du « Calendario do som »[8]. Cette ouvrage répertorie 366 compositions d’Hermeto Pascoal écrites en un an, une pour chaque jour. Les données sont enregistrées de façon à conserver les basses, le type d’accord, le rythme harmonique. Le but étant d’en tirer des outils de génération d’enchaînements harmoniques et d’harmonisation automatique.

2.3.1 Fonctions

Plusieurs fonctions ont été implémentées pour détecter les régularités harmoniques. Une fonction permet de rendre les données indépendantes de la tonalité et des notes pour n’utiliser que les intervalles. Des fonctions permettent de repérer les types d’accord, les enchaînements de 2, 3 ou 4 accords qui reviennent le plus souvent. Une fonction prend en entrée deux types d’accords et un nombre de temps n , et renvoie les différents enchaînements

harmoniques utilisés par le compositeur pour aller du premier accord au deuxième accord en n temps. Ces données étant indépendantes des notes, on peut ensuite générer les enchaînements harmoniques dans n'importe quelle tonalité.

2.3.2 Harmonisation dans le style d'Hermeto Pascoal

Une des fonctions implémentées fait le lien entre les 2 parties de ce stage. Elle permet d'harmoniser une mélodie ou un extrait de solo dans le style d'Hermeto Pascoal. Elle prend en entrée une mélodie sous la forme d'une liste de beats. Elle recherche dans la base de donnée les enchaînements harmoniques pouvant correspondre au niveau rythmique, elle les transpose ensuite dans les 12 tonalités. On applique la même méthode vue précédemment (cf. 3). Cependant, la fonction de calcul des poids d'un accord par rapport aux notes d'un beat utilise une nouvelle liste d'accords de référence. Celle-ci est composée de tous les accords rencontrés dans les morceaux intégrés. Les notes caractérisant ces accords sont les notes indiquées par Hermeto Pascoal dans le chiffrage des accords. Il utilise en effet une notation relativement riche qui nous permet de déduire les notes les plus importantes des accords. On calcule pour chaque enchaînement la somme des poids $\text{accord1} / \text{beat1}$, $\text{accord2} / \text{beat2}$, ... Elle renvoie finalement l'enchaînement harmonique ayant obtenu le poids le plus élevé.

Résultats

Les résultats obtenus au niveau de l'harmonisation automatique sont tout à fait corrects et on remarque immédiatement à l'écoute le style harmonique d'Hermeto Pascoal. La fonction est cependant assez lente à l'exécution.

2.3.3 Harmonisation dans le style d'Hermeto Pascoal avec indication sur l'accord précédent

On souhaiterait pouvoir harmoniser une séquence mélodique « par partie », c'est à dire utiliser la fonction d'harmonisation sur des parties de la mélodie. Il faut cependant maintenir une cohérence entre les résultats harmoniques de fin et de début de partie. On implémente pour cela une fonction qui prend en argument en plus de la mélodie le dernier «beat» harmonique précédent la mélodie. La recherche dans le corpus est alors restreinte à un

enchaînement harmonique tel que le précédent «beat» harmonique soit du même type, puis la transposition se fait de façon à rester dans la bonne tonalité.

On programme ainsi une autre fonction prenant en entrée un solo, sa pulsation associée et un nombre n . Elle harmonise dans le style d'Hermeto Pascoal en appelant la fonction précédente tous les n beats, tout en conservant une cohérence entre chaque partie grâce au dernier accord du segment précédent passé en paramètre.

Chapitre 3

Futures directions

A long terme le but de ce projet est de pouvoir utiliser ces différentes fonctions dans un contexte musical, dans la musique « live » plus particulièrement. En effet, le logiciel OMax crée un « clone » à partir d’une improvisation d’un musicien, donnant lieu à une interaction entre le musicien et la machine. La reconnaissance automatique d’accords peut permettre d’harmoniser automatiquement des figures musicales qui sont ensuite réinjectées pendant la performance, de générer des accompagnements harmoniquement cohérents avec l’improvisation. On peut imaginer également que la reconnaissance harmonique, particulièrement la fonction abordée dans , serve à se situer dans une structure telle qu’une grille pour pouvoir déclencher des évènements non seulement harmoniques mais aussi rythmiques (des « mises en place ») ou non musicaux ...

L’harmonisation automatique dans le style d’Hermeto Pascoal pourrait également permettre de générer des accompagnements, d’harmoniser des motifs mélodiques mais cette fois-ci dans le style du compositeur. Il est intéressant de noter que ces fonctions pourraient s’adapter au style d’un autre musicien, il suffirait d’incorporer un nouveau corpus. Celui-ci pourrait d’ailleurs être choisi par le musicien qui improviserait avec le logiciel.

Pour pouvoir réaliser ces différents traitements en temps-réel, il faudrait d’abord simplifier l’utilisation de ces fonctions, et dans certains cas les optimiser.

Conclusion

Bibliographie

- [1] E. Chew. Towards a mathematical model of tonality. Master's thesis, Operations Research Center, MIT, Cambridge, USA, 2000.
- [2] E. Gómez. Tonal description of music audio signals. Master's thesis, University Pompeu Fabra, Barcelona Spain, 2006.
- [3] C. Harte and M. Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the Audio Engineering Society*, Madrid, Spain, 2005.
- [4] P.R. Illescas, D. Rizo, and J.M. Iñesta. Harmonic, melodic, and functional automatic analysis. *Proceedings of the International Computer Music Conference (ICMC)*, pages 165–168, 2007.
- [5] K. Lee and M. Stanley. A unified system for chord transcription and key extraction using hidden markov models. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Vienna Austria, 2007.
- [6] L.C.L Neto. The experimental music of hermeto pascoal and group (1981 – 1993) : Conception and language. Master's thesis, UNIVERSIDADE DO RIO DE JANEIRO, Brazil, 1999.
- [7] Bryan Pardo and William P. Birmingham. Algorithms for chordal analysis. *Computer Music Journal*, 26(2) :27–49, Summer 2002.
- [8] H. Pascoal. *Calendário do som (2^a ed.)*. Senac São Paulo : Instituto Cultural Itaú, 2004.
- [9] T. Rocher, M. Robine, and P. Hanna. Amélioration des méthodes d'estimation d'accords et de tonalité depuis une représentation musicale symbolique. In *Journée d'Informatique Musicale*, 2009.
- [10] M.J. Steedman. A generative grammar for jazz chord sequences, 1984.
- [11] D. Temperley. The cognition of basic musical structures. *MIT Press*, 1999.