

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 aaaaa « 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 pulsation» (les notes MIDI situées entre deux mêmes pulsations sont regroupées) pré-existant dans le 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	9
1.2.1 Historique	9
2 Réalisation	13
2.1 Reconnaissance automatique d'accord	13
2.1.1 Algorithme de Pardo et Birmingham	13
2.1.2 Adaptation de l'algorithme et réalisation	15
2.2 Caler un extrait d'une improvisation sur une grille d'accord connue	22
2.3 Utilisation de l'harmonie d'Hermeto Pascoal	25
2.3.1 Etude de la musique d'Hermeto Pascoal et intégration dans l'environnement OpenMusic	26
2.3.2 Fonctions d'étude et de génération	29
2.3.3 Harmonisation dans le style d'Hermeto Pascoal	29
2.3.4 Harmonisation dans le style d'Hermeto Pascoal avec indication sur l'accord précédent	30
3 Futures directions	33
Conclusion	35

Table des figures

1.1	Architecture du système OMax	10
1.2	Exemple d'objet «beat»	12
2.1	Segmentation (source [9])	16
2.2	Exemple	19
2.3	Grille originale d'Israel	20
2.4	Première grille du solo de Bill Evans	21
2.5	Résultats pour la première grille du solo de Bill Evans	21
2.6	Extrait du solo de Bill Evans	24
2.7	Affichage de la grille harmonique décalée	25
2.8	Composition d'Hermeto Pascoal extraite du « Calendário do Som »	27
2.9	Exemple de notation d'accords	28
2.10	Exemple de grille harmonique dans l'environnement Open Music	28
2.11	Extrait mélodique avec harmonie originale	30
2.12	Extrait mélodique réharmoniser dans le style d'Hermeto Pascoal	30
2.13	Extrait mélodique réharmoniser dans le style d'Hermeto Pascoal par partie de 4 temps	31
2.14	Réharmonisation de Round Midnight dans le style d'Hermeto Pascoal par partie de 4 temps	32

Liste des algorithmes

1	Algorithme de reconnaissance automatique d'accord	17
2	Algorithme de calcul du poids d'un accord par rapport à un beat	20
3	Algorithme permettant de recalculer l'extrait d'une mélodie sur une grille d'accord	23

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 [8, 6]. 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 [5].

Dans le domaine de la musique symbolique, on citera le « modèle en spirale » de Chew [4] qui permet de modéliser notes, accords, et enchaînements d'accords. En plus de la tonalité, celui-ci peut estimer la suite d'accords d'un morceau en utilisant une fenêtre d'analyse où les éléments de la spirale sont étudiés grâce à des calculs géométriques. Les tests principaux ont été faits sur

des fugues de Bach. 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. [7]. 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 et 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.

Cette dernière approche est intéressante et aurait pu servir d'appui pour le stage mais elle n'est pas vraiment adaptable à une application « temps réel » car l'analyse du morceau ne peut se faire à la volée. On doit connaître la fin du morceau pour commencer l'analyse.

Rocher et al. [10] 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 [9] 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. C'est ce dernier algorithme qui a été choisi pour le projet car ces caractéristiques nous assuraient que l'on pouvait l'adapter à une situation « temps réel ». On détaillera

plus loin le principe de l'algorithme.

1.2 OMax

Le système OMax génère des improvisations en temps-réel. 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).

1.2.1 Historique

- 1998-1999

Gérard Assayag et Shlomo Dubnov publient leurs travaux sur la simulation stylistique : [1], [2].

À l'époque, il ne s'agit pas d'improvisation, mais de simulation stylistique réalisée en dehors de préoccupations de temps réel.

- juin 2001

Une première interface pour l'improvisation permettant le contrôle temps réel de procédures Open Music est réalisée directement en Lisp par Carlos Agon. Elle sert à implémenter des grammaires harmoniques développées par Marc Chemillier (GREYC, Caen, U.M.R. CNRS 6072), et est présentée aux JIM de Bourges en 2001 : [3]. Carlos Agon propose d'abandonner l'interface Lisp et de tenter un contrôle temps réel grâce à un protocole de communication Max / Open Music (d'où le nom OMax).

- 19 décembre 2001

Une boucle d'interaction Max / Open Music est écrite par Gérard Assayag. Marc Chemillier l'utilise pour réaliser au GREYC (Caen) durant l'année 2002 un premier prototype d'OMax basé sur les grammaires harmoniques qu'il étudie, et sur un système rudimentaires de génération de phrases improvisées.

- 31 octobre 2002

Marc Chemillier présente ce prototype à Bernard Lubat à Uzeste. Celui-ci manifeste son intérêt pour le projet et accepte de participer à des expériences pour développer l'interface du logiciel.

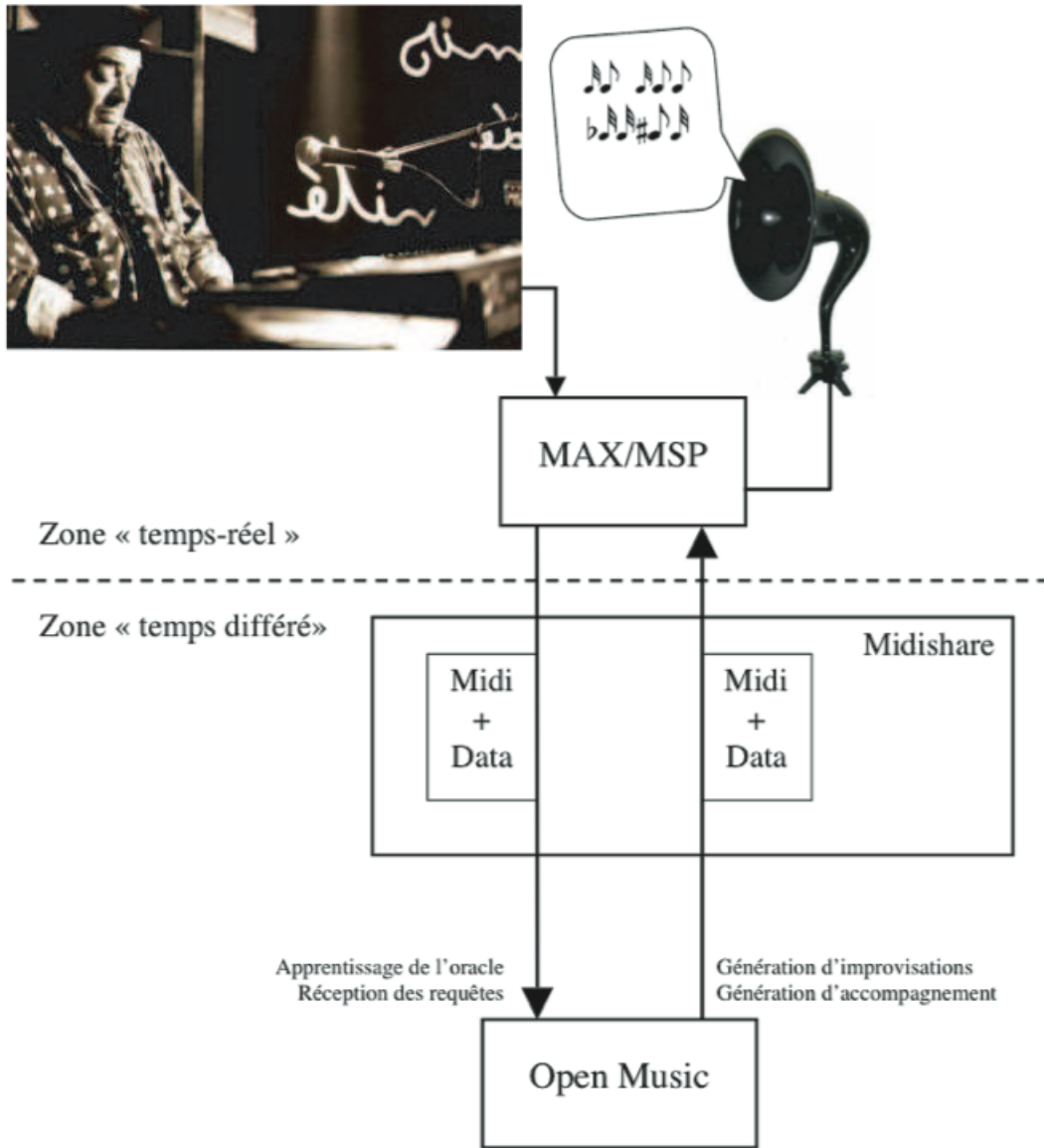


FIGURE 1.1 – Architecture du système OMax

- 2003-2004

Le modèle de l'oracle est intégré par Gérard Assayag dans OMax pour générer en temps réel des improvisations libres ("free"), et il l'étend au contexte d'une pulsation régulière et d'un cadre harmonique en définissant un objet Beat. Désormais OMax fonctionne selon deux modes possibles : "free" (sans connaissances musicales préalables) ou "beat" (avec prise en compte de la pulsation et de l'harmonie).

- 4 avril 2003

Première séance d'expérimentation de Bernard Lubat avec la nouvelle version du logiciel OMax intégrant les deux modes ("free" et "beat"), à Uzeste. Les premières missions à Uzeste nécessaires pour ces expérimentations sont financées par le GREYC (Caen).

- septembre 2003- septembre 2005

Marc Chemillier est en délégation CNRS auprès de l'Ircam pour continuer les recherches sur le logiciel. Georges Bloch (Université de Strasbourg) rejoint l'équipe de développement d'OMax et conçoit une extension du modèle de l'oracle pour traiter l'audio grâce à la mise au point d'un "pitch tracker" spécifique.

- juin 2004

Un financement est obtenu pour ce projet, et plus particulièrement la réalisation d'une animation multimédia présentant les différents modes d'interaction, dans le cadre d'un appel d'offre du Ministère de la recherche "Création de produits de médiation scientifique sur Internet".

- octobre 2004

Un grand workshop sur l'improvisation et l'ordinateur est organisé dans le cadre de Résonance à l'Ircam.

- 2005-2007

OMax connaît des développements importants avec le recours à un protocole de communication OSC, grâce auquel Gérard Assayag réécrit entièrement la boucle d'interaction Max / Open Music, et l'intégration de l'audio et de la vidéo dans le processus de recombinaison réalisée par George Bloch, qui est en délégation CNRS à l'Ircam à partir de septembre 2007. Dans cette nou-

Chapitre 2

Réalisation

2.1 Reconnaissance automatique d'accord

Dans les précédents travaux, les accords dans les objets «beat» étaient :

- soit insérés manuellement sur le piano-roll d'un éditeur MIDI
- soit insérés lors de la captation de l'improvisation lorsque celle-ci était produite sur un accompagnement joué par OMax (donc avec des accords connus d'avance)

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 [9], 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.

note	$\in \text{ accord}$	$\notin \text{ accord}$
$\in \text{ segment}$	poids + = poids de la note *	poids - = poids de la note *
$\notin \text{ segment}$	poids - = 1	..

TABLE 2.1 – Calcul du poids d'un *accord* par rapport à un *segment*

* Le poids de la note est égal au nombre de fois où l'on rencontre la note dans un des segments minimaux qui composent le segment en question.

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 simple de l'algorithme. On parcourt l'ensemble des segments minimaux un à un. À chaque étape on ne considère qu'un seul noeud c'est-à-dire un 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.

Détaillons ici la première étape.

Le premier segment minimal est composé des notes Do, Mi et Do. Le poids maximal (2) trouvé parmi tous les accords, correspond à l'accord de Do majeur composé des notes Do, Mi et Sol. (même score pour La mineur). En effet, les trois notes du segment sont dans l'accord donc 3 points. Il n'y a pas de note Sol dans le segment donc -1. Ce qui nous fait un poids de :

$$3 - 1 = 2$$

Le deuxième segment a obtenu un poids maximal de 0.

Si on fusionne ces deux segments on forme un nouveau segment composé des notes Do, Mi, Do, Ré, où les deux premières ont une valeur double car elles sont présentes sur les deux segments minimaux. L'accord de Do majeur obtient le meilleur résultat. Les notes Do et Mi appartiennent à l'accord donc :

$$2 + 2 + 1 = 5$$

La note Ré n'appartient pas à l'accord donc -1. Il n'y a pas de note Sol dans le segment donc -1. Ce qui nous fait un poids de :

$$5 - 1 - 1 = 3$$

Ce poids étant supérieur à l'addition des poids pris séparément (2 + 0), on ne considère plus que le segment « fusionné ».

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 réalisés pour deux raisons principales :

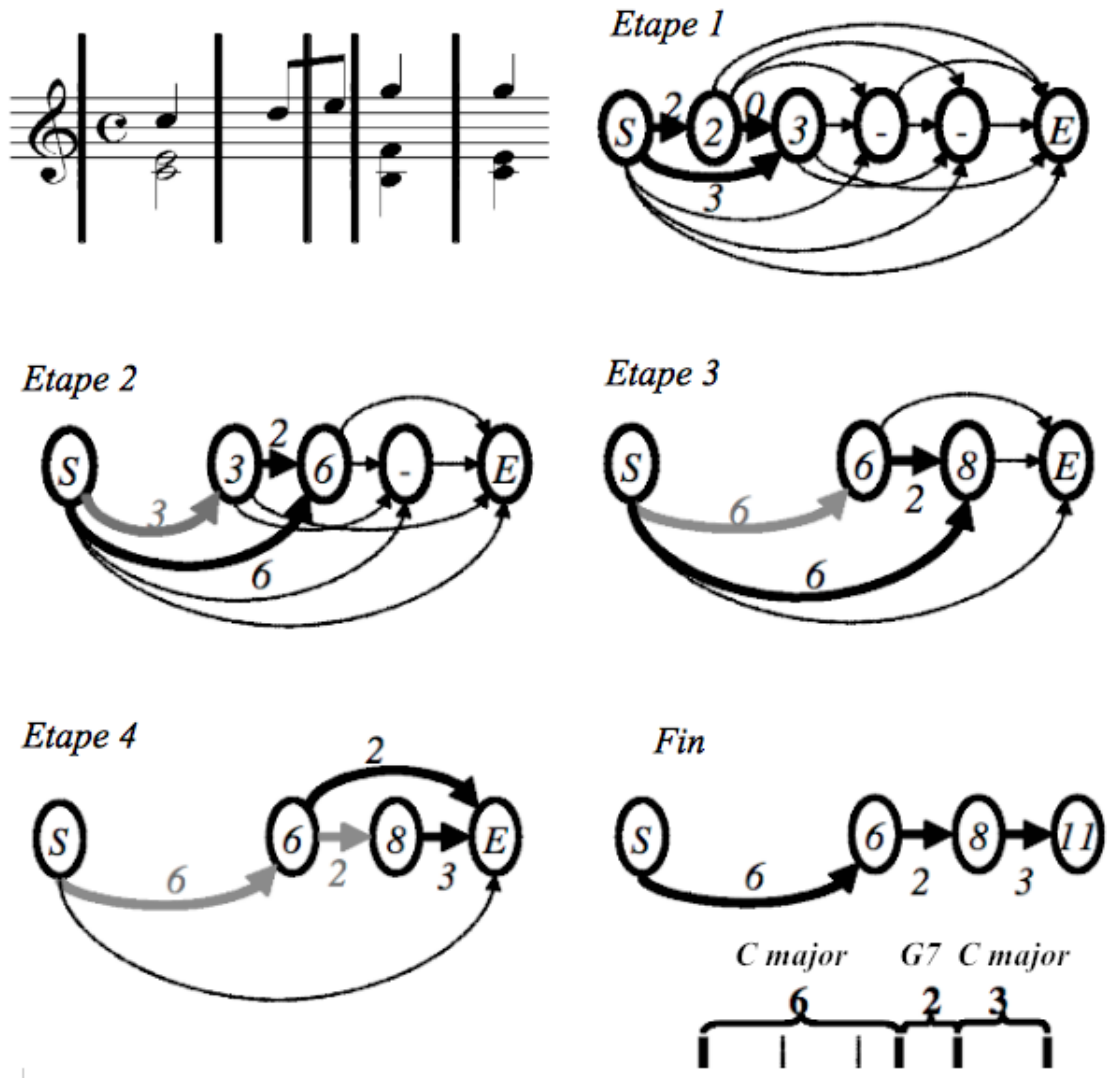


FIGURE 2.1 – Segmentation (source [9])

- le fait que l'algorithme soit à priori destiné à une analyse harmonique classique et non jazz.
- l'utilisation des objets «beat» préexistants dans OMax.

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 un temps rythmique, 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.

Algorithme 1 Algorithme de reconnaissance automatique d'accord

ENTRÉES: *beats* : liste de liste de quintuplet (fichier MIDI où les notes sont regroupées par beat)

SORTIES: *beats* avec les labels harmoniques associés à chaque beat
pour tout *b* dans *beats* **faire**

calculer puis trier les poids de tous les accords par rapport à *b*

fin pour

pour tout *segment* (équivalent au beats en premier lieu) **faire**

pmsprec ← poids max du segment précédent

pms ← poids max du *segment*

pm2s ← poids max des 2 segments fusionnés

si (*pmsprec* + *pms* ≤ *pm2s*) **alors**

on fusionne les 2 segments en 1 et on lui associe le label de poids max

finsi

fin pour

Retourner *beats* avec les labels harmoniques associés à chaque beat.

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 accord déjà présent du point de vue des intervalles à un renversement près. D'autre part on sait que dans le jazz les extensions des accords sont utilisés 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.

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

note	\in accord	\notin accord
\in beat	poids + = durée de la note	poids - = durée de la note
\notin beat	poids - = durée minimum des notes du beat	..

TABLE 2.2 – Calcul du poids d'un *accord* par rapport à un *beat*



FIGURE 2.2 – Exemple

Vème ...).

Exemple

Etudions un exemple de 4 temps (figure 2.1.2). On obtient les poids suivants pour chaque beat pris séparément :

beat 1 : -330 pour plusieurs accords

beat 2 : -328 pour plusieurs accords

beat 3 : 445 pour plusieurs accords

beat 4 : 562 pour plusieurs accords

Et obtient un poids de 2125 lorsqu'il fusionne les 4 beats pour les accords Dm7, F maj7.

Algorithme 2 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 ← 0

outcount ← 0

pour tout *n* dans *beat* **faire**

si ptich de *n* ∈ *accord*

incount += durée de *n*

sinon

outcount += durée de *n*

fin

fin boucle

missingcount ← nombre de notes de l'accord non présentes dans le beat
* min des durées

Retourner *incount* − (*outcount* + *missingcount*).

```
;Grille d'ISRAEL
;-----
;( d m7 d m7 d m7 d m7 | d m7 d m7 d m7 d m7 | d m7 d m7 d m7 d m7 | d alt d alt d alt d alt |)
;-----
;( g m7 g m7 g m7 g m7 | g m7 g m7 g m7 g m7 | f maj7 f maj7 f maj7 f maj7 | bb maj7 bb maj7 bb maj7 bb maj7 |)
;-----
;( e h-dim7 e h-dim7 e h-dim7 e h-dim7 | a 7b9 a 7b9 a 7b9 a 7b9 | d m7 d m7 d m7 d m7 | a 7 a 7 a 7 a 7 |)
;-----
```

FIGURE 2.3 – Grille originale d'Israel

Résultats

Israel est un morceau de John Carisi. Elle est composé de 12 mesures de 4 temps, soit 48 « beat ».

En effectuant les tests sur cinq grilles du solo de Bill Evans sur ce morceau (main droite + main gauche), on obtient environ 25% d'accords identiques à ceux de la grille originale, sur les 240 « beats ».

La figure 2.1.2 nous montre la grille harmonique (simplifiée) qui permet d'analyser nos résultats. La figure 2.1.2 nous montre la première grille du solo de Bill Evans. La figure 2.1.2 nous montre l'affichage des résultats sur cette extrait.

On remarque cependant qu'il n'y a jamais de résultat absurde au niveau

The image shows a piano accompaniment for a 12-measure solo. The music is in 4/4 time and consists of three systems of two staves each (treble and bass clef). The first system contains measures 1-3, the second system contains measures 4-8, and the third system contains measures 9-12. The melody is primarily in the right hand, while the left hand provides harmonic support with chords and moving lines.

FIGURE 2.4 – Première grille du solo de Bill Evans

```

OM > ;-----
OM > ( f maj7 f maj7 f maj7 f maj7 | f maj7 f maj7 g alt g alt | g alt c# alt g 7 g 7 | d alt d alt d alt d alt |)
OM > ;-----
OM > ( g m7 g m7 g m7 g m7 | g m7 c 7 c 7 c 7 | f 7 bb maj7 bb maj7 bb maj7 | bb maj7 bb maj7 bb maj7 bb maj7 |)
OM > ;-----
OM > ( e h-dim7 e h-dim7 e h-dim7 g h-dim7 | g h-dim7 g h-dim7 g h-dim7 a alt | c# alt e 7b9 e 7b9 e 7b9 | e 7b9 a alt a alt a alt |)
OM > ;-----

```

FIGURE 2.5 – Résultats pour la première grille du solo de Bill Evans

harmonique. Sur les 5 grilles du solo les principales différences sont les suivantes :

29 fois un accord C# alt (VIIème mode de la gamme mineure mélodique ascendante de D) à la place d'un accord D m7

16 fois un accord F maj7 (IVème mode de la gamme majeure de C) à la place d'un accord D m7 (IIème mode de la gamme majeure de C)

16 fois un accord E 7b9 (similaire à un accord D m7b6) à la place d'un accord D m7

13 fois un accord G demi-diminué (IVème mode de la gamme mineure harmonique de D) à la place d'un accord A 7b9 (Vème mode de la gamme mineure harmonique de D)

8 fois un accord Bb maj7 (IVème mode de la gamme majeure de F) à la place d'un accord F maj7 (Ier mode de la gamme majeure de F)

7 fois un accord Gm7 (IIème mode de la gamme majeure de F ou VIème mode de la gamme majeure de Bb) à la place d'un accord Bb maj7 (IVème mode de la gamme majeure de F ou Ier mode de la gamme majeure de Bb)

On voit que dans les principaux types d'erreurs, l'accord trouvé est dans la même gamme que l'accord attendu.

Parfois, l'erreur peut être 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 d'accord 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'élu.

Algorithme 3 Algorithme permettant de recalculer l'extrait d'une mélodie sur une grille d'accord

ENTRÉES: *beats* : liste de liste de quintuplet (fichier MIDI où les notes sont regroupées par beat) , une grille cyclique d'accord

SORTIES: la grille cyclique où le 1er accord correspond avec le début de *beats*

scoreMax ← -999999

pour *i* de 1 à longueur de la grille **faire**

score ← 0

pour *j* de 1 à longueur de *beats* **faire**

score ← *score* + poids du jeme accord par rapport au jeme beat

fin pour

si *scoreMax* < *score* **alors**

scoreMax ← *score*

resultat ← grille

finsi

grille ← *grilledcalde1accord*

fin pour

Retourner *resultat*.

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.



FIGURE 2.6 – Extrait du solo de Bill Evans

La figure 2.2 nous montre l'exemple sur lequel les résultats de la figure 2.2 ont été trouvés. On y voit la grille décalé de 6 mesures. Cela correspond avec le fait que l'extrait commence à la mesure numéro 7.

Caler un extrait d'une improvisation sur une grille d'accord parmi une liste grille

Une fonction supplémentaire a été implémentée afin de donner la possibilité à l'utilisateur de rentrer une liste grilles. Cela permet de reconnaître le morceau parmi un répertoire donné, et de se repérer dans cette grille. L'idée est la même que précédemment sauf que l'on choisit l'enchaînement harmo-

```

OM > ;-----
OM > ( f maj7 f maj7 f maj7 f maj7 | bb maj7 bb maj7 bb maj7 bb maj7 | e h-dim7 e h-dim7 e h-dim7 e h-dim7 | a 7b9 a 7b9 a 7b9 a 7b9
OM > ;-----
OM > ( d m7 d m7 d m7 d m7 | a 7 a 7 a 7 a 7 | d m7 d m7 d m7 d m7 | d m7 d m7 d m7 d m7 |)
OM > ;-----
OM > ( d m7 d m7 d m7 d m7 | d alt d alt d alt d alt | g m7 g m7 g m7 g m7 | g m7 g m7 g m7 g m7 |)
OM > ;-----

```

FIGURE 2.7 – Affichage de la grille harmonique décalée

nique qui a obtenu le plus gros score parmi les meilleurs scores de chaque grille.

2.3 Utilisation de l’harmonie d’Hermeto Pascoal

Le logiciel OMax permet de générer des improvisations à partir des données jouées par un musicien improvisateur. C’est notamment grâce à un automate nommé « l’oracle des facteurs » que le logiciel réinjecte ces données de façon cohérente. Il crée en quelque sorte un « clone » du musicien qui donne lieu à une réelle interaction entre le musicien et la machine.

Notre travail consiste à implémenter des enrichissements harmoniques qui viendront compléter les improvisations calculées par l’oracle. Il se situe donc dans la suite des précédentes expériences réalisées par M. Chemillier [?], [?], avec le modèle de "substitution" de la grammaire de Steedman [?]. Cette grammaire est définie sous la forme de règles de substitutions issues des techniques inventées par les Jazzmen «BeBbop» dans les années 40.

Pour développer ce travail d’enrichissement harmonique nous avons choisi de nous référer au compositeur et multi-instrumentiste brésilien Hermeto Pascoal.

Il est né le 22 juin 1936 à Arapiraca, dans l’État d’Alagoas, au Nord-Est du Brésil. Bien qu’il revendique un fort attachement à la musique traditionnelle brésilienne, il est une figure marquante du jazz. Il commença à être connu internationalement après que Miles Davis l’a invité en 1970 à participer à l’album enregistré en studio Live-Evil [?], dans lequel il joua plusieurs de ses propres compositions. Miles Davis a alors dit qu’Hermeto Pascoal était « le plus impressionnant musicien du monde. »

Hermeto Pascoal a une "oreille harmonique" particulièrement exception-

nelle, qui se traduit par sa capacité à imiter ou harmoniser n'importe quels sons qu'il entend dans la nature ([?], [?], [?], [?], [?] [?], [?]).

Compositeur prolifique, il s'est rendu célèbre par son projet « Calendário do Som » [?], dans lequel il a composé une chanson par jour, afin que tout le monde en ait une pour son anniversaire. Grâce au « Calendário do Som », on a la chance de disposer de plusieurs centaines de pièces, base de donnée unique en son genre. La figure 2.3 nous montre un exemple de composition telle qu'elle apparaît dans le recueil. L'harmonie du « Calendário do Som » a été notée de manière particulièrement précise, avec des chiffrages détaillés, plus complexes que ceux du Real Book par exemple, pour indiquer des sonorités précises voulues par le compositeur. Regardons la figure 2.3. Chaque accord indique le « voicing » exacte que l'on doit faire. Le premier accord E/B4568 par exemple correspond à : basse de Mi, Si puis quarte, quinte, sixte, et octave de B c'est-à-dire Mi, Fa#, Sol# et Si.

2.3.1 Etude de la musique d'Hermeto Pascoal et intégration dans l'environnement OpenMusic

Une analyse musicale de ses partitions, l'étude de documents de musiciens ayant joué avec lui, la consultation d'un rapport de thèse le concernant [?], et différents entretiens avec Jean-Pierre Cholleton (Professeur au conservatoire de Montreuil, spécialiste de la musique brésilienne), nous ont permis de mettre en évidence certaines de ses caractéristiques harmoniques. Par exemple l'utilisation quasi systématique d'accords mineur 4 7 9 en guise de second degré, l'utilisation répétée d'enchaînement d'accords parallèles ou d'accords identiques avec basse changeante.

Nous avons donc cherché à intégrer le style harmonique du compositeur au logiciel OMax d'une certaine façon. L'idée de départ était de repérer des règles de substitution à la manière de Steedman caractérisant le jeu harmonique d'Hermeto Pascoal à partir d'une analyse approfondie de sa musique, plus particulièrement en comparant l'harmonie de standards du jazz avec les réharmonisations d'Hermeto Pascoal de ces mêmes morceaux. Bien que nous ayons trouvé certaines régularités, nous n'avons pas pu en tirer des règles générales de substitution.

Nous avons donc décidé d'importer directement dans l'environnement Open Music un grand nombre de grilles issues du « Calendario do som ». Les données sont enregistrées de façon à conserver les basses, le type d'accord, le



FIGURE 2.9 – Exemple de notation d’accords

```
setf 30-de-junho '((a (- 9) 3)) ((g (/ 2 (mi)) 3)) ((d (- 9) 3)) ((c (/ 2 (mi)) 3))
((b (- 5-) 3)) ((ab (- 5-) 2)) (e (7 9-) 1)) ((a (- 9) 3)) ((b (- 5-) 2)) (e (7 9-) 1))
((a (/ 11 (-) 2)) (a (7 9-) 1)) ((d (- 9) 2)) (g (7 9-) 1)) ((c (7+) 2)) (f (7+) 1)) ((bb (7+) 2)) (e (7 9- 11+) 1))
((a (7+) 2)) (d (7+) 1)) ((g (7+) 3)) ((f# (- 4 7 9) 3)) ((f (7+) 2)) (ab (7+) 1))
((c (7+) 1)) (a (- 9) 1)) ((f (- 9) 2)) (bb (7 9 13) 1)) ((e (- 9) 3)) ((a (7 9- 13) 2)) (a (7 9-) 1))
((d (- 5-) 3)) ((g (4 7 9) 2)) (g (7 9-) 1)) ((g (4) 3)) ((b (- 5-) 2)) (e (7 9-) 1)))
```

FIGURE 2.10 – Exemple de grille harmonique dans l’environnement Open Music

rythme harmonique. Chaque composition correspond dans l’environnement Open Music à une liste de mesures, qui correspondent elles-mêmes à une liste d’accords. La figure 2.3.1 est la transformation dans Open Music de la composition de la figure 2.3. Un accord est représenté de la façon suivante :

(basse (“type d’accord”) (“nombre de temps”)).

Pour le type d’accord on reprend les notations du compositeur ; lorsqu’il s’agit d’une superposition d’accord on écrit :

(basse (/ “fondamentale de l’accord superposé” (“type d’accord”)) (“nombre de temps”)).

Les indications précises d’Hermeto Pascoal, nous a facilement permis de définir les notes de de tous les types d’accords rencontrés avec une basse de Do. Par exemple l’accord (4 7 9) est défini par (3600 5300 5800 6200) correspondance en midicent de (Do1 Fa2 Sib2 Ré3). On obtient tous les

types d'accords dans toutes les tonalités par transposition.

2.3.2 Fonctions d'étude et de génération

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.

2.3.3 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 la nouvelle liste d'accords. On calcule pour chaque enchaînement la somme des poids accord1 / beat1, accord2 / beat2, ... Elle renvoie finalement l'enchaînement harmonique ayant obtenu le poids le plus élevé.

Résultats

La figure 2.3.3 est extrait du solo de Art Pepper sur « You'd Be So Nice To Come Home » avec l'harmonie originale. La figure 2.3.3 nous montre la même phrase mélodique avec la réharmonisation trouvée par la fonction.

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.

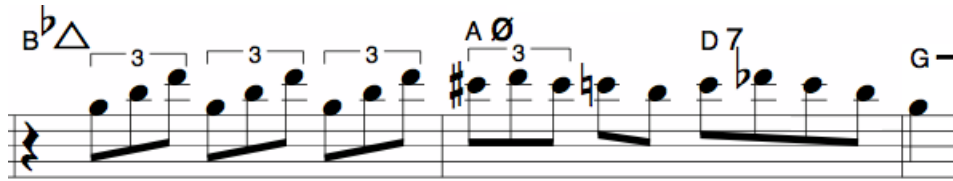


FIGURE 2.11 – Extrait mélodique avec harmonie originale



FIGURE 2.12 – Extrait mélodique réharmoniser dans le style d’Hermeto Pascoal

2.3.4 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.

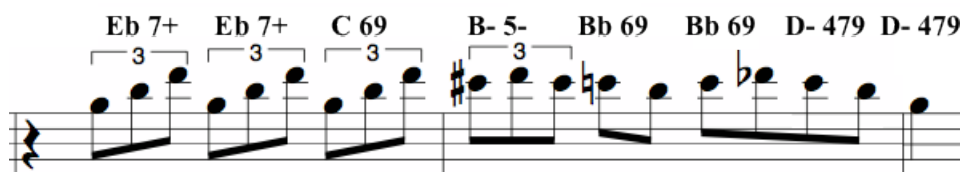


FIGURE 2.13 – Extrait mélodique réharmoniser dans le style d’Hermeto Pascoal par parties de 4 temps

Résultats

La figure 2.3.4 nous montre le même extrait du solo de Art Pepper mais cette fois-ci réharmoniser par parties de 4 « beats ».

Il existe plusieurs versions de réharmonisation sur le standard Round Midnight par Hermeto Pascoal lui-même. La figure 2.3.4 nous montre les résultats de la fonction par parties de 4 temps sur le début de ce standard. A l’écoute on note de réels similarités entre les deux réharmonisation.

Eb 69 F# 69 F 69 Eb69 Ab 7+ F- 479
 6

C# 7+ B 7+ Bb 7+ Ab 7+ G- 5- C 9+13- A 7+ D- 479
 8

G 7+ Db / E D / Eb D- 479 D- 479 D- / C
 10

B- 5- Bb 7+ D- 79 C- 79
 12

FIGURE 2.14 – Réharmonisation de Round Midnight dans le style d’Hermeto Pascoal par partie de 4 temps

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 ...

Une amélioration envisagée au niveau de la fonction qui permet de caler un solo sur une grille harmonique connue serait d’avoir la possibilité de reconnaître l’harmonie dans des morceaux où il est possible qu’une partie de celui-ci arrive de façon imprévue, un pont par exemple.

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 pourrait 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.

D’autre part, nous pouvons évoquer la possibilité d’utiliser l’oracle pour coder les enchaînements harmoniques. La fonction nommée « ImprovizeOnHarmGrid » déjà présente dans l’environnement prend en entrée :

- 1 oracle de « beats », c'est-à-dire d'objets associant 1 label harmonique + des données MIDI d'improvisation.

- 1 suite de labels harmoniques

Elle fabrique une séquence de midi data en "lisant" la suite de labels, et en concaténant les données MIDI correspondantes qu'elle trouve dans l'oracle. On pourrait faire la même chose, mais à l'envers, en construisant un oracle d'objets qui associent 1 "signature mélodique" (listes de notes caractéristiques d'un accord) + les labels qui peuvent harmoniser ces signatures. Ensuite, il ne reste plus qu'à calculer une suite de "signatures mélodiques" associée à la mélodie qu'on veut harmoniser. La fonction « ImprovizeOnHarm-Grid » prendrait alors en charge, automatiquement, le calcul d'une suite de labels qui convient. L'avantage est que l'on réutilise toute la mécanique de l'oracle sans la reprogrammer.

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. En effet l'exécution de certaines fonctions est parfois trop longue. L'efficacité pourrait être améliorée en utilisant un style de programmation plus adapté au langage fonctionnel (plus de récursif, moins d'itératif).

Conclusion

Ce stage s'est réalisé dans le cadre du projet ANR « IMPROTECH ». D'une durée de 4 mois, il a été consacré à l'étude et la réalisation d'outils visant à être intégrés au logiciel d'improvisation OMax. Deux parties étaient au départ clairement définies : la reconnaissance automatique d'accords, et l'étude de l'harmonie complexe de Hermeto Pascoal afin d'enrichir les phrases générés par le logiciel. Bien que nos idées concernant les deux problèmes n'aient aucun lien au départ, c'est tout naturellement que la résolution du premier nous permit d'avancer dans la deuxième partie. En effet, les fonctions les plus intéressantes réalisées au cours de ce stage utilisent toutes, d'une manière certes bien différentes, une fonction de calcul du poids d'un accord par rapport à une certaine mélodie, inspirée de l'algorithme de Pardo et Birmingham [9]. Nous avons finalement pu développer des outils permettant de se situer dans une structure harmonique, des fonctions de reconnaissance d'accords et d'harmonisation complexe dans le style du musicien Hermeto Pascoal, qui pourront facilement être intégrés au logiciel OMax.

Bibliographie

- [1] Gérard Assayag and Shlomo Dubnov. Universal classification applied to musical sequences. In *Proceedings of the ICMC : International Computer Music Conference*, Ann Arbor Michigan, 1998.
- [2] Gérard Assayag, Shlomo Dubnov, and Olivier Delerue. Guessing the composer's mind : Applying universal prediction to musical style. In *Proceedings of the ICMC : International Computer Music Conference*, pages 496–499, Beijing, 1999.
- [3] M. Chemillier. Improviser des séquences d'accords de jazz avec des grammaires formelles. In *Journée d'Informatique Musicale*, pages 121–126, Imeb, Bourges, 2001.
- [4] E. Chew. Towards a mathematical model of tonality. Master's thesis, Operations Research Center, MIT, Cambridge, USA, 2000.
- [5] E. Gómez. Tonal description of music audio signals. Master's thesis, University Pompeu Fabra, Barcelona Spain, 2006.
- [6] C. Harte and M. Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the Audio Engineering Society*, Madrid, Spain, 2005.
- [7] 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.
- [8] 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.
- [9] Bryan Pardo and William P. Birmingham. Algorithms for chordal analysis. *Computer Music Journal*, 26(2) :27–49, Summer 2002.

- [10] 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.
- [11] D. Temperley. The cognition of basic musical structures. *MIT Press*, 1999.