

平成 27 年度 卒業論文

クリシェによる
コード進行自動生成システム

函館工業高等専門学校 情報工学科 5 年

東海林研究室 江守南津郎

目次

第1章	序論	2
第1節	英文アブストラクト	2
第2節	研究目的	2
第3節	研究背景	3
第4節	開発環境	3
第2章	コード進行自動生成システムの開発	4
第1節	システムの概要	4
第2節	入力画面	5
第3節	メイン画面	5
第4節	キー解析	7
第5節	コード候補群生成	8
第6節	クリシエ推測	9
第7節	出力ファイル	11
第3章	出力例	12
第4章	評価実験	14
第1節	実験概要	14
第2節	結果	14
第3節	考察	15
第5章	結論	15
	参考文献	16

1 章 序論

第 1 節 英文アブストラクト

We sometimes see applications and software that create music automatically for a beginner without musical experience.

However the accompaniments have made by these systems are often commonplace. So the user seldom uses these systems repeatedly.

Therefore, the purpose of this study is to generate accompaniments that are not restricted by standard chord progression.

For that, this system adopts a technique called "cliché" for musicians to compose and arrange music.

At first, a user prepares MIDI data to which any melody has been inputted.

Then, the system analyzes the MIDI data. Finally, the system generates accompaniments that assign chords to the melody.

Also, the user is able to change at will the chord progressions.

We perform further experiments of the system against some users.

We also show the experimental results in the paper.

Key words : chord progression, automatic accompaniment generation, cliché

第 2 節 研究目的

作曲やアレンジの際に使用されるクリシェという技法を取り入れた、メロディーに対して動的に適切なコードを割り当てて伴奏を生成するシステムを作成する。

今回のシステムではまず初めに、ユーザは任意のメロディーを打ち込んだ MIDI データを用意する。その後、システムが MIDI データを解析し、メロディーに合ったコード進行を割り当てて伴奏を生成する。なお、ユーザによる自由な伴奏の変更も可能とする。

第3節 研究背景

近年、コンピュータの普及に伴い、作曲ソフトウェアが簡単に入手できるようになった。それにより楽器が演奏できない人や音楽初心者でも気軽に作曲を行うことが出来るようになった。しかし、作曲をするにあたり少なからず音楽的知識が必要となる。なかでもコード理論について初心者は難しく感じる人が多い。そのため音楽経験のない初心者のために自動作曲アプリやソフトがあるが、ありきたりな伴奏になることが多く、何度も利用しようと感じるものではない。

そこでこの研究ではコード理論に着目し、作曲やアレンジの際使用されるクリシェという技法を取り入れ、メロディーに対して動的に適切なコードを割り当てて伴奏を生成することにした。

第4節 開発環境

開発環境は以下のとおりである。

使用 PC Windows10 home 64bit version
 Intel Core 2 Duo CPU E7200 2.53GHz
 RAM 4.0GB

使用ソフト Visual Studio 2015
 Domino [1]

使用ライブラリ MIDIData ライブラリ [2]

使用言語 C++,C# [3]

2章 コード進行自動生成システムの開発

第1節 システムの概要

初めにユーザは任意のメロディーを打ち込んだ MIDI データを用意する。その後、システムが MIDI データを解析してメロディーにあったコード進行を割り当てて伴奏を生成し、生成された伴奏パートとメロディーパートを新規 MIDI データファイルとして出力する。出力される MIDI データは SMF (Standard MIDI Format) 1とし、分解能、テンポは元の MIDI データファイルの値を用いる。また、ユーザによる自由なコード進行に変更することも可能とする。

システムの処理の流れは図1のように、初めに MIDI データを読み込んで解析を行い、その後コード伴奏が付加された MIDI データファイルを出力する。MIDI データの解析処理内容は、大きく3つの処理から成り、それぞれキー解析部、コード候補群生成部、クリシェライン推測部と分けられる。

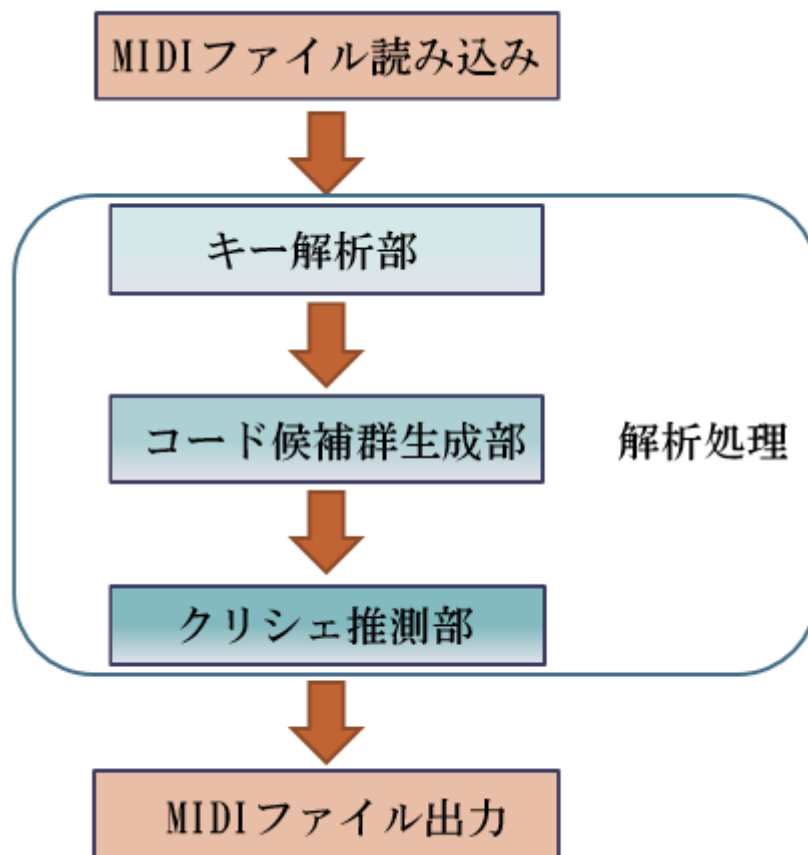


図1 システムの処理の流れ

第2節 入力画面

図2はシステム起動時の最初の画面である。ファイル参照ボタンで任意のMIDIデータを選択し、解析開始ボタンでMIDIデータの解析処理を行う。処理が終わるとMIDIデータを出力した後にメイン画面へ遷移し、コードの詳細なエディットを行うことができる。



図2 MIDIデータファイル読み込み画面

第3節 メイン画面

メイン画面ではキー解析部とコード候補群生成部での処理結果を表示している(図3)。キー解析部で決定したキーとキーの候補を右上のコンボボックスに示している。このキーはユーザによって自由に指定可能である。

またコード候補群生成部で生成された各小節でのコードの候補を中段のコンボボックスに順位とともに表示している。ユーザがコンボボックスのコード候補を選択し、変更を適用ボタンを押すと変更が適用されたMIDIデータファイルが再出力される。

さらにユーザがクリシェ推測ボタンを押すと、システムが最適だと判断したクリシェパターンを推測する(図4)。

下段にある「コードの種類指定」ではコードの種類やルート音をユーザが自由に指定できる。またPlay、StopボタンでMIDIの再生、停止を行う。

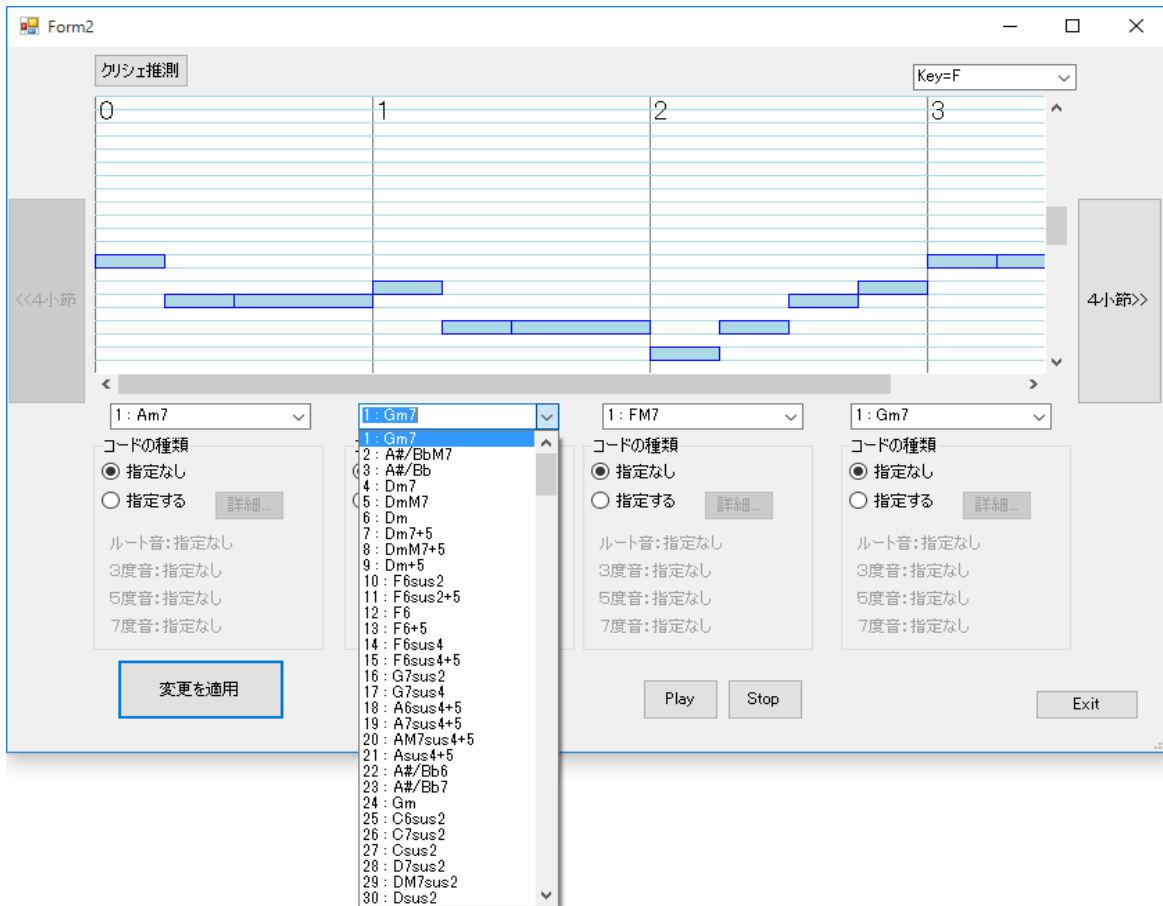


図3 メイン画面

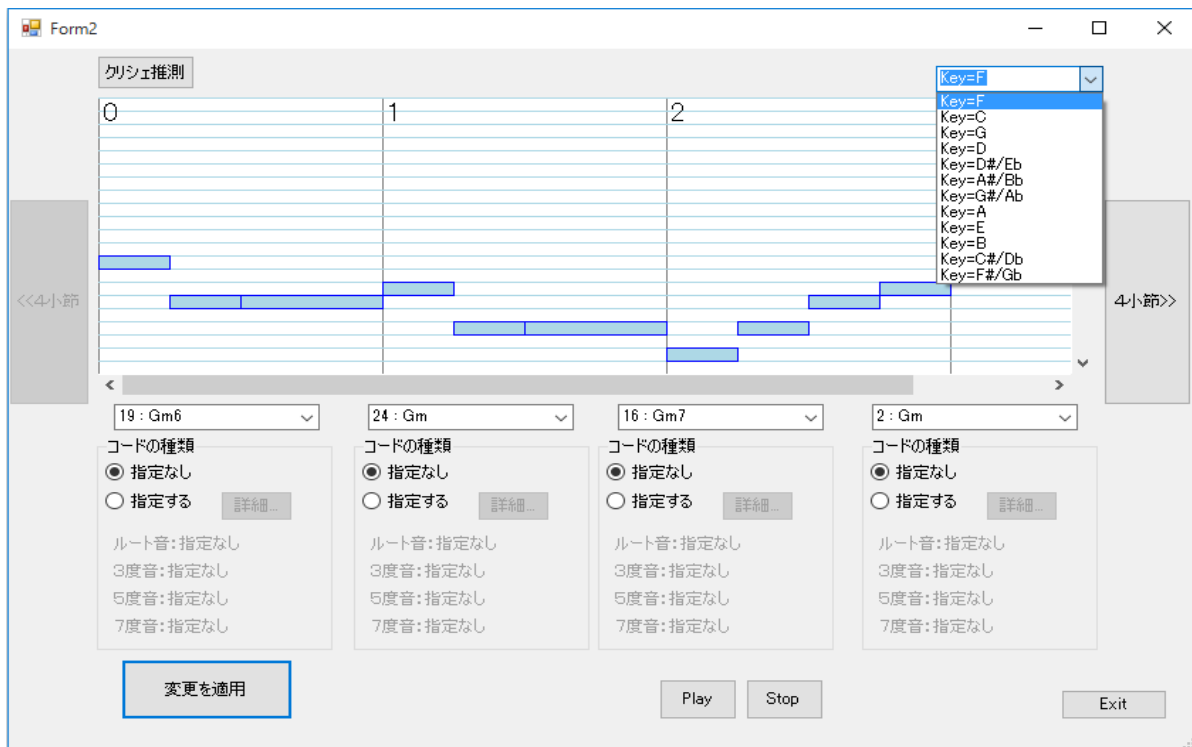


図4 クリシェ推測を行ったメイン画面

第4節 キー解析

キー (Key)とは曲のメロディーやコード、スケールの指標になるもので、キーが決まるとコードや使用できる音が決まる。Key が C のときは、主音の「ド」から「ドレミファソラシ」、Key が G のときは主音の「ソ」から「ソラシドレミファ#」といったように主に使われる音が決まっている[4]。

キー解析部では入力された MIDI データからメロディーを読み込み、メロディー全体で使われている各音の高さと長さを調べてデータ配列に格納する。その後メロディーの中に使用できない音と長さがどれだけ含まれているかを全てのキーに対して計算して順位を付ける。最終的に使用できない音を一番含まなかったキーをそのメロディーのキーとした。なお同順位が存在する場合、メロディーにキーの中心音がより多く含まれているキーを優先した。また全てのキーを可能性の高い順に順位付けし、ユーザに提示することでユーザによるキーの自由な変更も可能とした。なお今回は長調のみを想定している。

例として実際にキー解析を行った結果を図5、6に示す。図5は例として入力したメロディーである。



図5 入力メロディー

図6はこのメロディーに対するキーの解析結果である。各配列の添え字の0から11はドからシを示している。配列 gate は入力メロディー各音高の長さの総和であり、何の音がどれだけ鳴っているかを表している。今回は4分音符1つ分の長さを480に設定し、それを基準に8分音符では240、2分音符では960のように数値を決めた。例えば gate[2]は720なのでメロディー全体で「レ」の音が8分音符3つ分鳴っているということを意味する。

一方、配列 ScaleOutGate では各キーで使われない音がどれだけメロディーに含まれているかを示している。最も値が少ないものが最適なキーの候補である。なお最適なキー候補の値が同じ場合はキーの中心音がより多く含まれているものを最適なキーとした。


```

gate[0]=2640
gate[1]=0
gate[2]=720
gate[3]=0
gate[4]=0
gate[5]=240
gate[6]=0
gate[7]=1200
gate[8]=0
gate[9]=1680
gate[10]=240
gate[11]=0
ScaleOutGate[0]=240
ScaleOutGate[1]=3600
ScaleOutGate[2]=3120
ScaleOutGate[3]=1680
ScaleOutGate[4]=5040
ScaleOutGate[5]=0
ScaleOutGate[6]=6240
ScaleOutGate[7]=480
ScaleOutGate[8]=2400
ScaleOutGate[9]=4320
ScaleOutGate[10]=0
ScaleOutGate[11]=6480
第一候補
Key=5

```

図6 キー解析出力結果

第5節 コード候補群生成

コード(和音)は音名が異なる3つ以上の音から構成されており、メロディーを彩って曲全体のイメージを作る。またコード進行とは小節ごとにコードを割り当てたものである。楽器が伴奏を付ける際、コード進行を基に作成することが多い[5]。コード進行の例を図7に示す。

The image shows a musical score with two staves. The top staff is in treble clef with a key signature of one flat (Bb) and a 4/4 time signature. It contains a melody of quarter notes: G4, A4, B4, C5, B4, A4, G4, F4, E4, D4, C4. The bottom staff is in bass clef with the same key signature and time signature. It shows a chord progression with four measures, each containing a single chord: Am7, Gm7, FM7, and Gm7. A box labeled 'コード進行' (Chord Progression) is drawn around the bass staff. Above the treble staff, the chord names Am7, Gm7, FM7, and Gm7 are written above their respective measures.

図7 コード進行とメロディー

コード進行生成部では小節ごとにメロディーに合ったコードを推測する。本システムは3つの音から構成されるコードと4つの音から構成されるコード[6]を評価することでよりメロディーに合ったコード進行を生成する。評価方法としてメロディーの音の高さと長さに注目し、コードの構成音にメロディーの音が含まれる場合に長さに応じて加点した。またキー解析部で決定したキーを用いて、そのキーでよく使われる7つのコード(ダイアトニックコード)に対しても加点の評価を行った。さらに、決定したキーの音階に含まれない音をコードの構成音に含む場合は、1音ごとに減点を行った。これらの評価を全てのコードに行って評価点の高いものから順位付けをし、順位が高いものをよりメロディーに合ったコードとした。全ての小節に対してこれらの評価、順位付けを行いコードの候補群を生成する。生成されたコード群の中で一番順位の高いものを最適なコードとした。また評価点と同じものはのはドからシの順に並べられる。例として生成されたコードの候補群を図8に示す。

1: Am7	495	1: Gm7	495	1: FM7	375	1: Gm7	495
2: C6	480	2: A#/BbM7	495	2: D7sus2	360	2: Gm	495
3: C7	480	3: A#/Bb	495	3: D7sus2+5	360	3: Am7	495
4: C	480	4: Dm7	480	4: Dm7	360	4: C6sus2	480
5: Edim7	480	5: DmM7	480	5: Dm7+5	360	5: C7sus2	480
6: Em7-5	480	6: Dm	480	6: Em7+5	360	6: Csus2	480
7: EmM7-5	480	7: Dm7+5	480	7: E7sus4+5	360	7: C6	480
8: Edim	480	8: DmM7+5	480	8: F6sus2	360	8: C7	480
9: Em6+5	480	9: Dm+5	480	9: FM7sus2	360	9: C	480
10: Em7+5	480	10: F6sus2	480	10: F6	360	10: C6sus4	480
⋮		⋮		⋮		⋮	

図8 生成された候補群

第6節 クリシェ推測

クリシェとはコード進行のテクニックでよく使用される1つのテクニックであり、コード進行の中で特定の1つの音が半音または全音上下に動くことである。クリシェを用いることで単調なコード進行に変化を持たせることが可能である[7]。またクリシェにはいくつか種類があり、例えばコードの5度と7度の音を変化させるメロディッククリシェ、ルート音を変化させるベースクリシェ、2つの音を同時に変化させるハーモニッククリシェ等があるが、今回はメロディッククリシェに限定して実装を行った。クリシェを用いたコード進行の例を図9に示す。

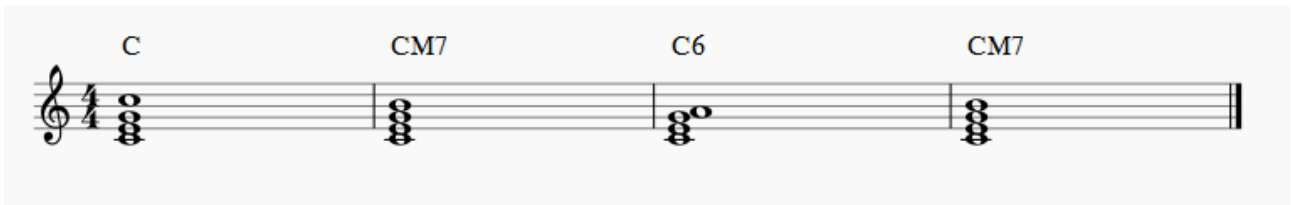


図9 クリシェを用いたコード進行

クリシェ推測部ではコード候補群生成部で順位付けを行ったコード群を用いてよりメロディーにあったクリシェパターンを推測する。具体的にはクリシェを開始する小節の全てのコード候補群から5度もしくは7度の音を全音か半音ずらしたコードが次の小節のコード候補群に含まれているかどうかを探索する。もし含まれている場合は、そのコードから同様に5度か7度の音を全音または半音ずらしたコードが次の小節のコード候補群にないか探索を行っていく。コードの遷移先の表を表1に示す。クリシェを終える小節まで来た場合、今まで遷移してきたコードとコードの順位の総和を1つのクリシェ候補とする。例として探索を行う様子を図10に示す。

一方、コードが含まれていない場合は別の遷移先の探索を行う。全てのパターンの探索を行いコードの順位の総和がより少ないものを最適なクリシェを用いたコード進行とする。

表1 5度もしくは7度の音を全音か半音ずらしたコードの遷移表

Gm→Gm-5	Gm→Gm+5	Gm→Gm6	Gm→GmM7
Gm→Gm7	Gm-5→Gm	Gm-5→Gm	Gm-5→Gm+5
Gm+5→Gm	Gm+5→Gm-5	Gm+5→Gm6	Gm+5→Gm7
Gm6→Gm+5	Gm6→Gm	Gm6→Gm7	Gm6→GmM7
Gm7→Gm6	Gm7→Gm+5	Gm7→GmM7	Gm7→Gm
GmM7→Gm7	GmM7→Gm6	GmM7→Gm	

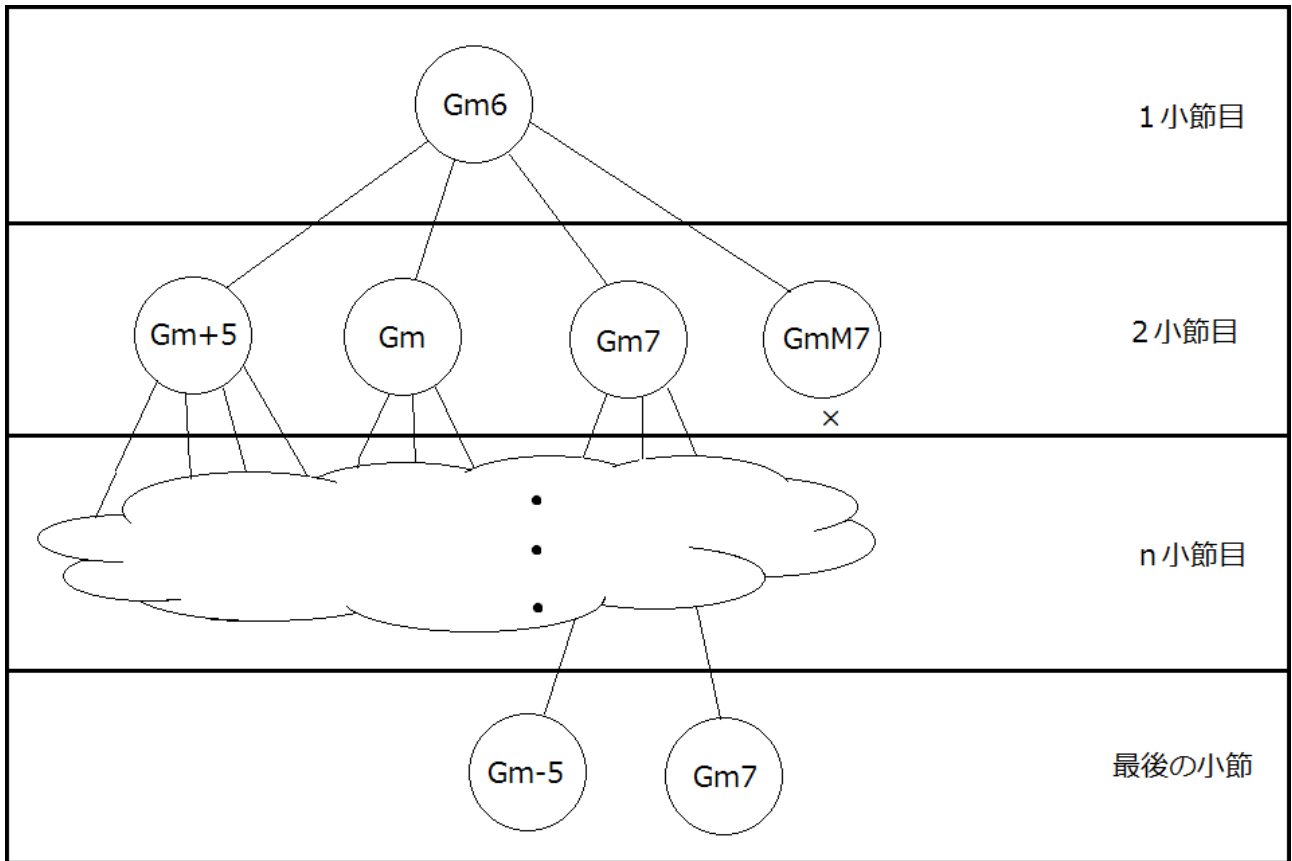


図10 クリシェパターンを探索する

第7節 出力ファイル

MIDIデータの解析を行い最適と判断したコード進行伴奏を付与したMIDIデータファイルを新規ファイルとして出力する。テンポや分解能は入力MIDIデータと同様とし、複数のMIDIトラックを使用するためSMF1 (Standard MIDI Format1)[8]を使用した。最終的な出力例を図11に示す。

♩ = 120

The musical notation shows a melody line in the treble clef and a bass line in the bass clef, both in 4/4 time. The tempo is marked as quarter note = 120. The melody consists of quarter notes, and the bass line consists of chords.

図11 最終出力例

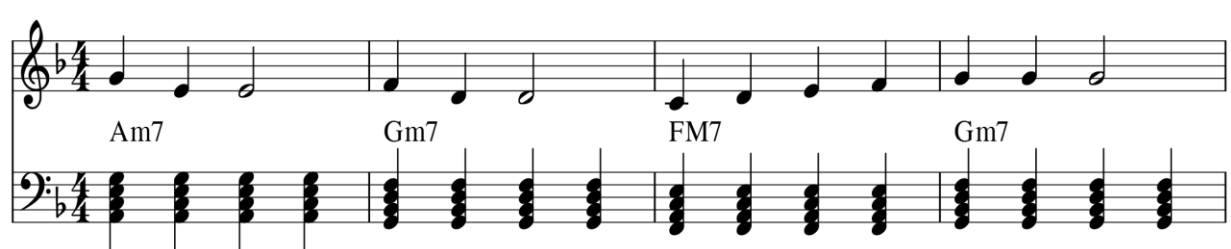
3章 出力例

ここでは実際にメロディーが記された MIDI データファイルを解析してコード進行を生成した例を挙げる。

例1 ちょうちょ

メロディーとして童謡のちょうちょ[9]の出だし4小節を用いる。図12にコード候補群生成で生成された最適なコード伴奏例を示す。

♩ = 120



Am7 Gm7 FM7 Gm7

図12 ちょうちょに割り当てられたコード伴奏

また最適なクリシェパターンを推測してメロディーに割り当てたコード伴奏例を図13に示す。

♩ = 120



Gm6 Gm Gm7 Gm

図13 ちょうちょに割り当てられたクリシェを用いたコード伴奏

例2 もみじ

メロディーとして童謡のもみじ[10]の12小節目から16小節目を用いる。図14にコード候補群生成で生成された最適なコード伴奏例を示す。

♩ = 120

C6sus2 FM7 C6sus2 Am7

図14 もみじに割り当てられたコード伴奏

また最適なクリシェパターンを推測してメロディーに割り当てたコード伴奏例を図15に示す。

♩ = 120

Am7 Am Am+5 Am7

図15 もみじに割り当てられたクリシェを用いたコード伴奏

4章 評価実験

第1節 実験概要

実際に本システムを用いて被験者からメロディーにコード伴奏を付けてもらいアンケートを行った。被験者は17～20歳で4名に行った。実施日は2016年2月24日(水)である。

被験者は、2種類の用意されたメロディーから1つ選んでもらい、メロディーにコード伴奏を付けてもらう。その後、出力されたクリシェを用いないコード伴奏と、クリシェを用いたコード伴奏を聞いてもらったあとにアンケートを取った。アンケートは5段階評価で高いほど良い。アンケートの内容は下記のとおりである。

- ・簡単にコード伴奏をつけられたか
- ・クリシェを用いないコード伴奏のクオリティはどうか
- ・クリシェを用いたコード伴奏のクオリティはどうか
- ・実際に使ってみたいか

第2節 結果

結果を表2に示す。

表2 アンケート結果

アンケート項目	1人目	2人目	3人目	4人目	平均
簡単にコード伴奏をつけられたか	4	5	4	5	4.50
クリシェを用いないコード伴奏のクオリティはどうか	3	3	4	3	3.25
クリシェを用いたコード伴奏のクオリティはどうか	5	4	4	4	4.25
実際に使ってみたいか	3	4	1	4	3.00

また、以下のコメントを得た。

- よりオーケストラ感が増した
- 推測時間が長い
- クリシェの無駄遣い
- 計算時間が長い、Web やモバイル上でなら使ってみたい

第3節 考察

「簡単にコード伴奏をつけられたか」は全員4以上であり、本システムで手軽に伴奏をつけることができたといえる。「クリシェを用いた伴奏」と「クリシェを用いない伴奏」のクオリティではクリシェを用いた伴奏の方が評価が若干高かったため、本研究の目的を達成しているといえる。

一方、「実際に使ってみたいかどうか」は評価が良くなかった。原因としてクリシェ推測の時間が長いことや Web などでも手軽に使えなかったことが考えられる。

5章 結論

本研究では作曲やアレンジの際に使用されるクリシェという技法を取り入れ、メロディーに対して動的に適切なコードを割り当てて伴奏を生成するシステムの作成を行った。更に実際にユーザに使用してもらいアンケートを行った結果、簡単にコード伴奏をつけることができるということが分かった。

今後の課題として、クリシェの推測アルゴリズムを改良して時間を短くしたり Web やモバイル上への移植を行いたい。

参考文献

[1]MIDI 音楽編集ソフト「Domino (ドミノ)」 | TAKABO SOFT

<http://takabosoft.com/domino>

[2] MIDI データ作成・編集用ライブラリ

<http://openmidiproject.osdn.jp/MIDIDataLibrary.html>

[3] 池谷京子 増田智明 国本温子 著, Visual C# 2008 逆引き大全(秀和システム版, 2009)

[4] キー(調性) / スケール(音階) / コード(和音)の関係 | 88notes / 実践! DTM で作曲する人のためのコード進行・音楽理論

<http://www.88notes.net/?p=134><http://www.88notes.net/?p=134>

[5] 成瀬正樹著, コード進行スタイルブック(RittorMusic, 2001)

[6] 篠田元一著, 実践コード・ワーク(RittorMusic, 2005)

[7] クリシェを使ったコード進行 | 88notes / 実践! DTM で作曲する人のためのコード進行・音楽理論

<http://www.88notes.net/?p=346>

[8] MIDI ファイルの基礎知識

<http://hp.vector.co.jp/authors/VA029289/midi1.html>

[9] 「ちょうちょう」の楽譜 / おとたま

<http://ototama.com/music/folksong/score.php?id=195>

[10] 畑中良助著, 小学生の音楽 4(株式会社 教育芸術者, 2009)