

E-01

動画をもとにした自動作曲のユーザ適応について User Adaptation of Automatic Composition based on Video Clips

山本 敏生† 宝珍 輝尚† 野宮 浩揮†
Toshiki Yamamoto Teruhisa Hochin Hiroki Nomiya

1. はじめに

近年、youtube をはじめとした動画共有サービスの台頭により、世界に向けて自作動画を手軽に公開できる環境が整ってきた。それによって、個人による動画製作の機会が増加し、同時に、動画に使用する BGM への需要が高まってきている。現在では、著作権フリーの音楽素材をダウンロードして使用するのが主流であるが、動画に合った音楽を探し出すのは容易ではない。また、著作権法で保護されている音楽を動画に付加して web 上に違法アップロードする者も多数存在し、大きな問題となっている。

これらの問題への解決策として、動画に合うような楽曲を自動的に生成する自動作曲システムを提案し、開発を進めている[1]。これにより、動画ごとにオリジナルの楽曲を生み出し、BGM として利用することが可能となる。

本システムの課題として、「ユーザ適応」が挙げられる。現状では、音楽心理学を参考にして音楽生成を行っているが、こうして生成された音楽がユーザの嗜好に合致するとは限らない。

そこで本論文では、この問題を解決するためにユーザ適応に関してこれまでに行ってきたシステムの変更について述べる。具体的には、マルコフ過程の一つであるマルコフ連鎖を用いたコード進行生成法と、曲の転換の閾値を操作する GUI について述べる。

以降、2.で関連研究について述べ、3.で自動作曲システムの概要とその問題点を示す。そして、4.でコード進行生成について述べ、5.で閾値操作について述べる。最後に 6.でまとめを行う。

2. 関連研究

マルコフ過程を用いた自動作曲の手法に関しては、過去に様々な研究がなされている。

- ・マルコフ過程を用いて音高の状態遷移を推測した上で、次音を決定する。[2]
- ・音符の隣接情報を音高と音価に分けて考え、音高は隣接情報によって構成されたマルコフ過程を使って生成し、音価は、各音価の一つ前の音価の終了位置(次の音価の開始位置)を隠れ状態とした隠れマルコフモデルによって生成する。[3]
- ・学習データをある作曲家の MIDI データ(縦に音階、横に時刻とし 2 次元の格子として扱う)とし、作曲家のスタイルや特徴をマルコフ確率場の特徴関数と重みで表現することで画像と同様に特徴の学習を行い、

この確率場から確率の高い標本をモンテカルロ法により得ることで自動作曲を行う。[4]

これらの研究には以下のような共通点がある。

- ・単音の旋律しか生成できない
- ・既存の譜面や曲データが必要となる

本研究では上記の問題を回避するために、音楽理論に基づいたコード進行をマルコフ連鎖で生成する手法を試みている。

GUI に関しては、絵をかくように音楽を作成できるソフト「Hyperscore」が既に開発されている[5][6]。ユーザが特別な音楽知識を持っていなくても、作曲することが可能である。本研究ではこのソフトを参考にして、後述する”場面転換の閾値”を直感的に操作できる GUI を開発している。

3. 自動作曲システム

3.1 概要

システムの動作を図 1 に示す。

初めに動画ファイルを読み込み、動画ファイルから全フレームをビットマップとして切り出した後、各フレームの色情報を抽出する。この色情報を用いて音楽情報を生成する。

動画の中で画面が大きく変化するタイミングを「場面転換」と見なし、場面転換してから、次に場面転換するまでの区間を、一つの「ブロック」とする。各ブロックに含まれるフレームの色情報をもとにして、ブロック毎に、コード進行・旋律・リズムといった音楽情報を決定していく。色情報と音楽情報の対応付けは、音楽心理学[7]を参考にして行っている。

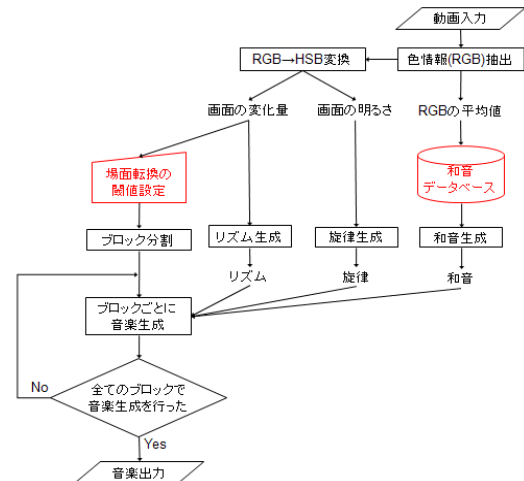


図1：自動作曲システムの動作

3.2 色情報とコード進行の対応付け

様々な研究より、長和音と暖色、短和音と寒色がそれぞれ対応していることが分かっている[1]。この性質を用いてコード進行を生成する。

基本的には、循環コードを用いて作曲を行う。4小節で1つのパターンを作り、それを繰り返して曲を構成していく。循環コードのパターンはブロックごとに変更される。

また、RGB成分の平均から、R成分とB成分の割合を求め、それをもとにメジャーコードとマイナーコードの割合を表1のように決定する。

表1：メジャーコードとマイナーコードの割合

Rの比率	メジャーコード：マイナーコード
80%以上	4：0
60%以上 80%未満	3：1
40%以上 60%未満	2：2
20%以上 40%未満	1：3
0%以上 20%未満	0：4

3.3 問題点

このシステムには以下のような問題点がある。

問題1：音楽情報は基本的に乱数によって生成されており、音楽理論に基づいていない。従って、不協和音が生成される可能性があり、また、曲調やジャンルの指定ができない。

問題2：このシステムでは場面転換の基準を、HSBの差を計算した値が閾値を上回った場合としているが、閾値の設定次第では、聴くに堪えない音楽が生成される可能性がある。例えば、画面の変化が激しい動画を読み込んだ場合、常に場面転換していると見なされて、最初から最後まで常にコード進行やリズムが変動する支離滅裂な曲が生成されたり、逆に、画面がほとんど変化しない動画を読み込んだ場合、全く場面転換していないと見なされて、最初から最後までコード進行やリズムが全く変化しない退屈な曲が生成されるといったことが起こりうる。

これらの問題を解決するために、コード進行の生成と、閾値設定について改善を行っている。

4. コード進行生成

3.3で述べた問題1に対応するため、コード進行をコード理論に基づいて生成するようにした。

4.1 コードの機能 [8]

コード進行を分析する際、そのコードが属する調における立場や役割を主に3種類にまとめて分類する。これらを機能またはファンクション(function)という。その名前はスケール上のI音とIV音とV音に由来し、これらをルートとするコードが、3種類の機能を代表している。以降はそれぞれ、T(トニック)、S(サブドミナント)、D(ドミナント)と表記する。

TSDの連結、つまりコード進行における機能のつながりは以下のようにまとめることができる。

- T→S→T
- T→D→T
- T→S→D→T

また、メジャー・キーのVI音は、キーにあまり影響を及ぼさずに半音下げる変化が可能である。メジャー・キーのSの中のVI音をbVI音に変化させた機能をSのマイナー、つまりサブドミナント・マイナーということがあり、SMと略記する。SMはメジャー・キーのS→Tの間に挟むのが一般的である。SMを用いたコード進行における機能のつながり方は以下ようになる。

- T→SM→T
- T→S→SM→T
- T→SM→D→T
- T→S→SM→D→T

これらをまとめると、コード機能の遷移図は図2のように描くことができる。各状態への遷移確率は等確率である。このようにコードを遷移させていくことで、コード進行の自動生成が可能になる。

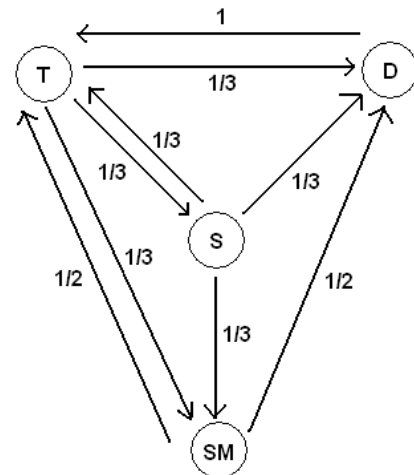


図2：コード機能の遷移図(図中の数値は遷移確率を表す)

4.2 コードデータベース

システムの実現にあたって、コードを一括して管理できるコードデータベースを作成した。

テーブルの構造を表2に示す。

表2：テーブルの構造

Field	Type	Key
degree_name	varchar(20)	PRI
function	varchar(2)	
substitution	tinyint(1)	
diatonic_scale	tinyint(1)	
diatonic_chord	tinyint(1)	
tension	tinyint(3)	

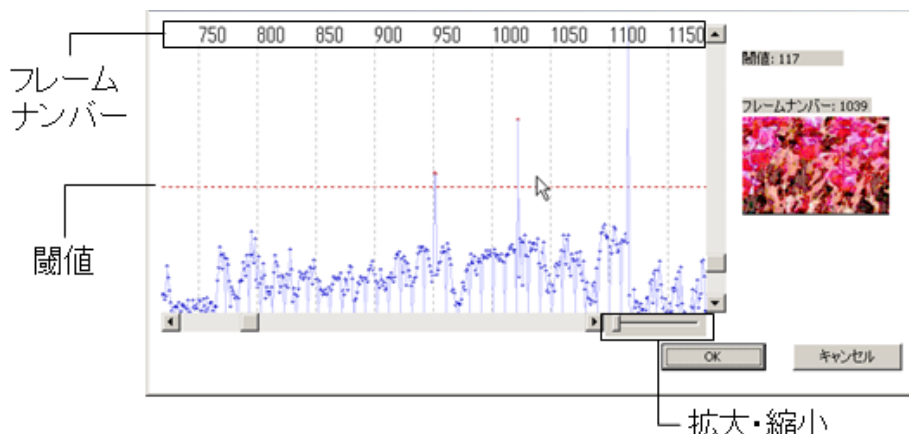


図 5 : 閾値操作ウィンドウ

フィールド名の”degree_name”はコードのディグリーネーム, ”function”はコードの機能, ”substitution”はコードが代理和音であるか否か, ”diatonic_scale”はコードがダイアトニックスケール上の音のみで構成されているか否か, ”diatonic_chord”はコードがダイアトニックコードか否か, ”tension”はテンションノートの数を, それぞれ表している。[8]

3.2 で述べた色情報との対応付けと, コード進行に関する音楽理論を用いて, コードデータベースからコードを取得し, コード進行を生成する。ブロックが切り替わるたびに, コードデータベースから 4 小節分のコードを取得する。

4.3 出力結果

風景が場面ごとに次々と切り替わる 2 分 50 秒の動画 sample.mpg を用意し, システムに読み込んで, ブロックごとにコード進行を生成した。結果の一部を図 4 に示す。小文字の”m”が付いているコードがマイナーコードである。

4.1 で挙げた規則通りにコード進行が生成されていることが分かる。

```

c:\Documents and Settings\Yamamoto\My Documents\Visual Studio 2008\Projects\FAC2\Debu
<Block 34> Red: 56.34% Blue: 43.66% Chord: I6->IIm7onV->VIm->V7
<Block 35> Red: 56.5171% Blue: 43.4829% Chord: I6->IIm->VIm->I1M7
<Block 36> Red: 56.5428% Blue: 43.4572% Chord: IM7->IIm7->I1Im7->V7sus4
<Block 37> Red: 56.0612% Blue: 43.9388% Chord: IM7->IIm7onV->I1Im7->IV6
<Block 38> Red: 60.6342% Blue: 39.3658% Chord: I6->IIm6->IVonV->I
<Block 39> Red: 49.5141% Blue: 50.4859% Chord: IM7->IIm6->VIm7->IVM7
<Block 40> Red: 65.6112% Blue: 34.3888% Chord: IM7->IIm6->V->I
<Block 41> Red: 35.5054% Blue: 64.4946% Chord: VIm->IIm->V7sus4->VIm7
<Block 42> Red: 65.494% Blue: 34.506% Chord: I->IIm7onV->I->IV
<Block 43> Red: 23.513% Blue: 76.487% Chord: VIm7->IIm6->V->I1Im
<Block 44> Red: 65.4162% Blue: 34.5838% Chord: IM7->IIm7->I->IV
<Block 45> Red: 13.5014% Blue: 86.4986% Chord: VIm->IIm7onV->VIm->I1m7
<Block 46> Red: 65.4307% Blue: 34.5693% Chord: I6->IIm6->V->I
<Block 47> Red: 5.02822% Blue: 94.9718% Chord: I1Im7->I1m7->I1Im7->I1m
<Block 48> Red: 65.5687% Blue: 34.4313% Chord: I->IIm6->IM7->V7sus4
<Block 49> Red: 19.255% Blue: 81.745% Chord: I1Im7->I1m7onV->I1Im7->I1m7onV
<Block 50> Red: 65.6239% Blue: 34.3761% Chord: I6->IIm6->V7sus4->I
<Block 51> Red: 54.0514% Blue: 45.9486% Chord: IM7->IIm->VIm7->IVM7
<Block 52> Red: 30.1343% Blue: 69.8657% Chord: I1Im->V7->I1Im7->IIm6
<Block 53> Red: 54.142% Blue: 45.858% Chord: I1Im7->IIm6->V->I1m
<Block 54> Red: 42.6609% Blue: 57.3391% Chord: VIm7->IIm7->IVonV->I
<Block 55> Red: 54.8161% Blue: 45.1839% Chord: VIm->IIm->I6->IVM7
<Block 56> Red: 53.2898% Blue: 46.7101% Chord: I1Im->IIm->I->IVM7
<Block 57> Red: 55.8945% Blue: 44.1055% Chord: I1Im->IIm->I->V7
<Block 58> Red: 63.6777% Blue: 36.3222% Chord: I->IV6->VIm7->V
<Block 59> Red: 56.1565% Blue: 43.8435% Chord: I1Im->IIm6->IM7->V
<Block 60> Red: 73.991% Blue: 26.009% Chord: I6->IVonV->I1Im7->IV6
<Block 61> Red: 56.2824% Blue: 43.7176% Chord: VIm7->IIm7->I6->IV
<Block 62> Red: 82.2857% Blue: 17.7143% Chord: I->IV->I6->IV6
<Block 63> Red: 56.2021% Blue: 43.7979% Chord: VIm->IIm->I6->IVonV
<Block 64> Red: 80.0919% Blue: 19.9081% Chord: IM7->V7->IM7->V7sus4
<Block 65> Red: 54.6114% Blue: 45.3886% Chord: I1Im->IIm6->IM7->V
<Block 66> Red: 75.3693% Blue: 24.6317% Chord: I6->V->I1Im->IVonV
<Block 67> Red: 52.5347% Blue: 47.4653% Chord: I1Im->IIm7onV->I->V7
<Block 68> Red: 70.6853% Blue: 29.3147% Chord: IM7->IVonV->VIm->IV
<Block 69> Red: 54.101% Blue: 45.899% Chord: VIm7->IIm6->V->I
<Block 70> Red: 65.7192% Blue: 34.2808% Chord: IM7->IV->I1m7onV->I6
<Block 71> Red: 58.8952% Blue: 41.1048% Chord: VIm7->IIm7onV->IM7->IVonV
<Block 72> Red: 44.2077% Blue: 55.7923% Chord: I1Im->IIm7onV->I6->IVonV

```

図 4 : 実行結果

5. 閾値の操作

3.3 で述べた問題 2 に対応するため, 閾値操作ウィンドウをシステムに追加し, ユーザが場面転換の閾値を自由に変更できるようにした。音楽生成の直前に, 図 5 に示すウィンドウが表示され, 閾値の操作が可能になる。

閾値操作ウィンドウの仕様は以下の通りである。

- ・ 波形は HSB 変化量を表している
- ・ 画面を横切る破線は閾値を表している
- ・ 画面上部の数字はフレーム数を表している
- ・ 画面右下のスライダーを操作することで, グラフの拡大・縮小ができる
- ・ グラフ上で破線をドラッグすることで, 閾値を操作できる
- ・ 画面右側にはマウスカーソルが指すフレームナンバーに対応するフレームの画像が表示される

これを用いて閾値変更を行なった結果を図 6, 7 に示す。動画 sample.mpg を読み込んだ後, コード進行生成の前に閾値を変更することでブロック数が変化していることが分かる。

```

c:\Documents and Settings\Yamamoto\My Documents\Visual Studio 2008\Projects\FAC2\Debu
Clip Length: 170.534
Frame Rate: 29.97
Number of Frames: 5110
threshold: 250
<Block 1> Red: 54.6143% Blue: 45.3857% Chord: I1Im7->I1m7onV->I6->IV
<Block 2> Red: 61.9622% Blue: 38.0378% Chord: I6->IV6->I1Im7->IV6
<Block 3> Red: 49.9936% Blue: 50.0064% Chord: I1Im->I1m7onV->I6->V7
<Block 4> Red: 45.3212% Blue: 54.6788% Chord: VIm7->IIm->V7->I
<Block 5> Red: 50.0017% Blue: 49.9983% Chord: I1Im7->I1m7->V7sus4->I6
<Block 6> Red: 53.6005% Blue: 46.3995% Chord: VIm7->IIm->IVonV->I6
<Block 7> Red: 50% Blue: 50% Chord: VIm->I1m7onV->IM7->IV
<Block 8> Red: 55.752% Blue: 44.248% Chord: VIm->IIm->V7->I
<Block 9> Red: 65.412% Blue: 34.588% Chord: IM7->I1m7->VIm->IVonV
<Block 10> Red: 13.5014% Blue: 86.4986% Chord: VIm7->I1m7onV->VIm7->I1m7onV
<Block 11> Red: 65.4307% Blue: 34.5693% Chord: I->VIm7->I1m7onV->I6
<Block 12> Red: 5.02822% Blue: 94.9718% Chord: I1Im->I1m7onV->I1Im7->I1m
<Block 13> Red: 65.5687% Blue: 34.4313% Chord: I6->IV6->I1Im7->V7
<Block 14> Red: 18.255% Blue: 81.745% Chord: VIm7->IIm7->VIm->IIm6
<Block 15> Red: 57.4637% Blue: 42.5363% Chord: VIm->V->I6->IV6->I1m7
<Block 16> Red: 30.1343% Blue: 69.8657% Chord: I1Im7->IVonV->I1m7->I1m
<Block 17> Red: 54.142% Blue: 45.858% Chord: I1Im->IV->I->I1m6
<Block 18> Red: 42.6609% Blue: 57.3391% Chord: VIm->V->IM7->I1m
<Block 19> Red: 54.8161% Blue: 45.1839% Chord: VIm7->IV6->V->VIm7
<Block 20> Red: 53.2898% Blue: 46.7101% Chord: I1Im7->IVM7->V7->VIm7
<Block 21> Red: 55.8945% Blue: 44.1055% Chord: I1Im7->IVonV->I6->I1m6
<Block 22> Red: 63.6777% Blue: 36.3222% Chord: IM7->V7->I1Im7->V7
<Block 23> Red: 56.1565% Blue: 43.8435% Chord: VIm->IV6->V->I1m7
<Block 24> Red: 73.991% Blue: 26.009% Chord: I6->IV->I1Im->V7
<Block 25> Red: 56.2824% Blue: 43.7176% Chord: I1Im7->V->I6->I1m
<Block 26> Red: 82.2857% Blue: 17.7143% Chord: I->IV->V7sus4->I
<Block 27> Red: 56.2021% Blue: 43.7979% Chord: I1Im7->IV6->V7sus4->VIm
<Block 28> Red: 80.0919% Blue: 19.9081% Chord: IM7->V7->IM7->V
<Block 29> Red: 54.6114% Blue: 45.3886% Chord: I1Im->IV->V7->I1Im
<Block 30> Red: 75.3693% Blue: 24.6317% Chord: I->V->I1m6->IV6
<Block 31> Red: 52.5347% Blue: 47.4653% Chord: I6->IVonV->I1m7->I1m
<Block 32> Red: 44.8981% Blue: 55.1019% Chord: IM7->IVonV->I1Im7->I1m6

```

図 6 : 閾値 = 250 の場合

```

c:\Documents and Settings\Yamamoto\My Documents\Visual Studio 2008\Projects\AC2\Debu...
Clip Length: 170.534
Frame Rate: 29.97
Number of Frames: 5110
threshold: 300
<Block 1> Red: 54.8627% Blue: 45.1373% Chord: I6->IIIm7->IIIm7onV->IM7
<Block 2> Red: 50% Blue: 50% Chord: IM7->IIIm7->VIIm7->V7sus4
<Block 3> Red: 53.2968% Blue: 46.7032% Chord: I->IIIm->IIIm7onV->I6
<Block 4> Red: 30.1343% Blue: 69.8657% Chord: IM7->IIIm7onV->IIIm7->IIIm7
<Block 5> Red: 54.142% Blue: 45.858% Chord: IM7->IIIm7->VIIm7->IV
<Block 6> Red: 42.8609% Blue: 57.3391% Chord: I->IIIm7onV->IIIm7->IVM7
<Block 7> Red: 54.8161% Blue: 45.1839% Chord: I6->IIIm6->IIIm->IV
<Block 8> Red: 53.2899% Blue: 46.7101% Chord: I->IIIm->IIIm->V
<Block 9> Red: 55.8945% Blue: 44.1055% Chord: IIIm7->IIIm7onV->IM7->IVM7
<Block 10> Red: 63.6777% Blue: 36.3223% Chord: VIIm->V7sus4->I->IVonV
<Block 11> Red: 56.1565% Blue: 43.8435% Chord: VIIm->IIIm->IVonV->I6
<Block 12> Red: 73.991% Blue: 26.009% Chord: VIIm->IV6->IVonV->I
<Block 13> Red: 56.2324% Blue: 43.7676% Chord: IIIm->IIIm7onV->I->V7
<Block 14> Red: 82.2357% Blue: 17.7643% Chord: IM7->V->IM7->I6
<Block 15> Red: 56.2021% Blue: 43.7979% Chord: VIIm7->IIIm7->IVonV->I6
<Block 16> Red: 45.7357% Blue: 54.2643% Chord: IIIm7->IIIm7onV->I6->V7

```

図 7： 閾値 = 300 の場合

閾値を 250 に設定した場合、ブロックの数は 32 となり、閾値を 300 に設定した場合は、ブロックの数が 16 となっている。これより、閾値の上下によって場面転換挿入のタイミングが増減していることが分かる。

6. おわりに

今回は、動画をもとにした自動作曲システムについて、ユーザ適応に関する改善を行った。

今後の課題としては、コード進行や場面転換の閾値だけでなく、旋律やリズムについてもユーザ適応を考慮したシステムを構築することが挙げられる。ユーザに音楽的なパラメータを操作させる場合、特別な音楽知識を要さずとも、システムを扱えるようにする工夫が必要である。

参考文献

- [1] 山本敏生, 宝珍 輝尚, 野宮 浩揮: “動画をもとにした自動作曲”, 平成 21 年度情報処理学会関西支部大会論文集, C-22 (2009)
- [2] 三重野芳典, 椎塚久雄: “自動作曲システム”, 人文科学とコンピュータ 46-4, 音楽情報科学 35-4, pp.25-30 (2000)
- [3] 耿霽, 池田剛, 乾伸雄, 小谷善行: “隠れマルコフモデルを用いた曲構造を持つ音符列の生成”, 音楽情報科学 49-6, pp.31-36 (2003)
- [4] 住田浩之, 林朗: “マルコフ確率場を用いた自動作曲”, 2008-MUS-74(27), 2008-SLP-70 (27), pp.151-156 (2008)
- [5] “HYPERSCORE.COM”, <http://www.hyperscore.com/>
- [6] “Hyperscore (Versions 1.0 - 4.3.2, 2000-2006)”, <http://web.media.mit.edu/~mary/hyperscore.html>
- [7] 谷口 高士: “音は心の中で音楽になる 音楽心理学への招待”, 北大路書房 (2000)
- [8] 林知行: “標準ポピュラー・コード理論”, 株式会社シンコーミュージック・エンタテイメント (2006)