

# ギター演奏に特化した 自動譜めくりシステムのための楽譜追跡手法

鈴木涼太郎<sup>1</sup> 野田夏子<sup>1</sup>

**概要**：近年、音楽分野でも楽譜の閲覧のために電子媒体が利用されるようになり、演奏者の演奏に追従して自動で楽譜をめくる自動譜めくりの技術が研究開発されている。自動譜めくりのために必要になる技術のひとつが楽譜追跡技術であるが、実際の演奏では弾き誤りや弾き飛ばし、テンポの揺らぎなどの楽譜とは異なる演奏が発生するため、これらの不確定要素を含む演奏に対しても正確に楽譜追跡を行う必要がある。しかし、多声楽器での演奏による多重音の音響信号を入力とする場合には、既存手法では正確な楽譜追跡が難しい。そこで本稿では、対象をギターのコード演奏に絞り、楽曲とギター演奏の特性を考慮することで、様々な不確定要素を含む実際の演奏に対しても対応できる楽譜追跡手法を提案するとともに、提案手法に基づいて開発したギター演奏のための自動譜めくりシステムのプロトタイプについて報告する。また、この開発経験から、こうした音楽演奏支援のソフトウェア開発における課題についても考察する。

## 1. はじめに

近年では音楽分野でも楽譜の閲覧のために電子媒体が利用されるようになり、それに伴って電子楽譜上で利用される、演奏者の補助を目的としたシステムの研究が行われている。自動譜めくりはその一つであり、自動譜めくりに要求される要素技術が楽譜追跡である。自動譜めくりの利用場面として想定される練習時の演奏では、弾き誤りや弾き飛ばし、テンポの揺らぎなどの、楽譜とは異なる要素が発生するため、これらの不確定要素を含む演奏にも追従可能なシステムが求められる。音響信号を入力とする場合は、単旋律の演奏を扱う実用レベルのシステムが開発されているものの、多声楽器での演奏による多重音の音響信号を扱うシステムには様々な課題が存在する。

バンドで演奏されるギターは多くの場合、主旋律を演奏する「リード」と呼ばれるパートと伴奏を演奏する「バックギター」と呼ばれるパートに分かれており、特にバックギターではコード演奏が行われることが多く、多重音での演奏が中心である。よって、ギター演奏における楽譜追跡では、多重音の追跡が必要となる。

本稿では、ギターのコード演奏のための自動譜めくりの特化し、様々な不確定要素にも追従可能な楽譜追跡手法を提案する。また、提案手法を用いた自動譜めくりシステムのプロトタイプ開発について述べる。一般的な手法としての精度向上を目指すのではなく、適用範囲を絞ることで、譜めくりの対象となる楽譜や演奏の特徴を明らかにすることができ、実用に耐えると期待される手法を提案することができた。一方で、人の演奏という自由度の高い入力を扱うことによる評価の難しさなどの開発上の問題にも直面した。本稿ではこうした点に関しても考察する。

## 2. 自動譜めくりに必要なとされる技術

### 2.1 音高推定

音高推定は自動譜めくりと自動伴奏に加え、自動採譜の分野でも研究が行われている技術である。自動採譜とは音響信号からコンピュータによって楽譜を作成する技術であり、人が演奏に利用するための一般的な形式の楽譜を作成するためだけではなく、コンピュータによる楽曲の理解や楽曲分析を行うことにも応用される。そのため、高い精度の音高推定が要求される。自動採譜の分野では機械学習を組み合わせることで、高い精度での音高推定が可能となってきており、部分的ではあるが、ピアノ譜において実用レベルの精度が実現されている[1]。一方で、自動譜めくりや自動伴奏の分野では、あらかじめ演奏される楽譜がわかっているため、自動採譜ほどの精度は要求されないが、入力された音高信号に対して逐次的に演奏音を特定していくことが必要となるため、リアルタイムでの音高推定が求められる。リアルタイムでの音高推定は単旋律の演奏においては高い精度での音高推定が可能となっている[2]が、ギターやピアノなどの多声楽器においてはリアルタイムで処理可能な和音数に制限があり、実際の演奏においては精度が低下してしまう[3]、といった課題が存在する。

### 2.2 楽譜追跡

楽譜追跡は自動伴奏などの分野で盛んに研究が行われており、演奏に追従して楽譜位置の推定を行う技術である。楽譜追跡には、演奏中の誤りが予測できない場合や、テンポの変動、楽譜上にない表現などが多数ある場合には精度が低くなってしまうという課題が存在する。また、自動譜めくりの分野では楽譜をめくるタイミングさえわかればよく、自動伴奏と比べて正確な演奏位置を特定する必要がないため、その特徴を生かした手法[4]も提案されている。

<sup>1</sup> 芝浦工業大学  
Shibaura Institute of Technology

楽譜追跡において、既存研究では MIDI 信号と音響信号の 2 通りが主に入力として用いられている。MIDI 信号を入力とする研究では高い精度での追跡が実現されている [5]が、MIDI 入力を持つ楽器にしか対応することができない。一方で、音響信号を入力とした研究では、多声音の演奏は音響信号が複雑になり、演奏音の特定が困難であるため十分な精度は実現できていない。また、既存研究での手法では、演奏に大きな誤りがないことが前提とされているなどの制約がある、類似したフレーズが繰り返し演奏されるような特定の特性を持つ楽曲では追跡制度が大幅に減少してしまう [6]、演奏に含まれる和音数や楽曲の長さが増加すると計算量が膨大になりリアルタイムでの動作ができなくなってしまう [7]、といった課題が存在する。

### 3. 関連研究

#### 3.1 コード推定の基本手法

コード推定は、主に自動採譜の分野で要求される要素技術である [8]。ある区間でのクロマベクトルを分析することで演奏されたコードの推定を行う。クロマベクトルとは、音響信号において音階周辺の周波数をすべて 12 音階に分類し、それぞれの音階ごとに振幅を足し合わせることで、オクターブ違いを含めた各音階の振幅強度を求めたものである。また、クロマベクトルを描画したものをクロマグラムという。図 1 は実際のギター演奏のクロマグラムである。横軸が時間、縦軸が音階を表し、色の明るさで振幅強度の大きさを表している。

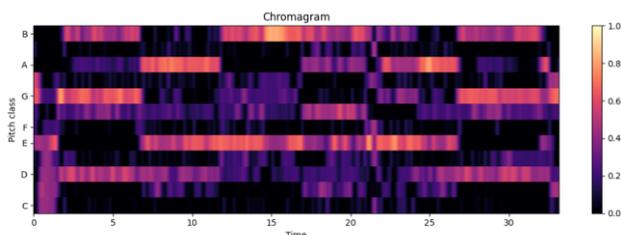


図 1 クロマグラムの例

このクロマグラムを分析することで、音響信号に含まれる音階を特定することができる。そして、一定区間の演奏音におけるクロマベクトルと、ベクトル化したそれぞれのコードの構成音の存在比でのコサイン類似度が最も大きいものが推定コードとなる。

#### 3.2 An On-Line Algorithm for Real-Time Accompaniment

Dannenberg らは、自動伴奏を目的とした単旋律の楽譜追跡手法として、DP マッチング (動的計画法) を用いた手法を提案した [9]。

Dannenberg らは演奏と楽譜との「ベストマッチ」を見つけることを単旋律の楽譜追跡における課題とし、「ベストマッチ」を「演奏と楽譜における最長の共通部分列」と定

義している。図 2 はベストマッチの例を示したものである。

performance:	A	G	E	D	G	B	C
				/			
best match:	A	G	E	G	B	C	
				/			
score:	A	G	E	G	A	B	C

図 2 ベストマッチの例 ([1]より引用)

ベストマッチを見つけるために、Dannenberg らは行を楽譜、列を演奏とした整数行列を利用し、動的計画法を用いてそれぞれの値を決定している。なお、 $r$  行  $c$  列の整数は  $r$  番目までの楽譜と  $c$  番目までの演奏を比較したときの共通部分列の長さを表している。このアルゴリズムにより共通部分列の長さを求め、 $(r, c)$  の値が  $(r-1, c-1)$ 、 $(r-1, c)$ 、 $(r, c-1)$  の値よりも大きいときをベストマッチとする。

また、この手法では、楽曲が長くなった場合に計算量が大幅に増加してしまう。そのため、楽譜と演奏の比較を行う範囲に窓を設定し、前の列でベストマッチがあった場合は窓の中心を一致した行の次の行に設定、ない場合は窓の中心を現在の中心の行の次の行に設定することで、計算量の削減を行う。この窓の大きさは少なくとも連続する間違いの 2 倍以上にする必要があり、想定される演奏に応じて窓の大きさを設定する。さらに、この手法では、最初の音で弾き間違いが発生すると演奏位置の大幅なスキップが発生してしまうという課題が存在する。例えば、BGEFCD という楽譜に対し、FBG のような、演奏の頭に余分な音が入ってしまった場合、楽譜の 4 番目に F が存在するため、それ以前の BGE は弾き飛ばしであると判断されてしまう。この課題を解決するために、Dannenberg らはさらに以下の 3 つの改善手法を提案している。

- ・ 演奏ごとの窓の中心の変化量を 2 行までに制限する
- ・ 楽譜の弾き飛ばしをした際の一致数にペナルティをかける
- ・ 候補が複数あるとき、先に演奏されるものをベストマッチとする

#### 3.3 自動譜めぐりシステムのための多重音に対する音高・音源数の高速推定法

安部らは自動譜めぐりシステムのための音高と音源数の同時推定法の提案と、処理時間の比較を行っている [3]。

安部らの手法では、演奏音の短時間フーリエ変換により得られたパワースペクトルのうち一定以上のパワーを持つピーク位置を抽出し、倍音に対応することが明らかなものを削除していくことで得られる基本周波数から演奏音と音数の推定を行う。従来の手法では和音数の増加に伴って処理時間の増加がみられていたが、このアルゴリズムを用いることで 5 和音までは処理時間が増大することなく推定を

行うことができる事が示されている。しかし一方で、実際の演奏では音源数や音高に誤りが多く見られることが課題とされている。

### 3.4 音響信号の振幅を用いるオーケストラ楽曲の楽譜追跡

三浦らは、オーケストラでの演奏を対象とした、音響信号を入力とする楽譜追跡手法を提案している[6]。この研究では、演奏がリズム、音高も含め楽譜に記述されている情報をほぼ忠実に再現していることを前提としている。また、オーケストラでは同時に多くの楽器が演奏され、周波数解析によって音高を検出することが非常に困難であるため、この手法では音高情報を用いず、演奏音の振幅に注目し、オーケストラ全体のリズムによって生まれる振幅変化と楽譜から推測される演奏音の振幅変化でのマッチングをすることで楽譜追跡を行う。

### 3.5 楽譜の自動譜めくりシステムのための演奏位置推定方法

小倉らは、自動譜めくりでは自動伴奏のような音符レベルでの演奏位置推定を行う必要はない、という点に着目し、2段階のマッチングによる小節単位での演奏位置推定を行っている[4]。第一段階では全く同じ演奏が繰り返されない場合において、小区間ごとに楽譜を区切った場合に、それぞれ特有の音高出現頻度のパターンが存在するとし、一定時間内の演奏での音階出現頻度とあらかじめ作成した区間ごとの音高出現頻度を比較することで、演奏位置の候補を求める。さらに第2段階では旋律的な情報を考慮し、第一段階で求めた候補区間をさらに小区間に分割し、小区間での音の出現数を比較することで区間全体での音の出現数の変化を調べる。そして演奏にも同様の処理を行い、小区間同士の出現数の差を比較し、また演奏の連続性を考慮してマッチングに補正を行うことでさらなる演奏位置の絞り込みを行う。

## 4. ギター演奏のための自動譜めくりの要件

本稿で対象とするギター演奏のための自動譜めくりにおいては、ギターからの音響信号からリアルタイムで演奏音が特定できることと、実際のギター演奏時に発生しうる弾き誤りや弾き飛ばしといった楽譜と異なる要素を含む演奏にも追従できることが必要である。

キーボードのような、MIDI 出力を持つ楽器であれば、MIDI 出力を利用することで音高や発音時刻などの情報を直接出力することが可能であるが、ギターのような一般的に MIDI 出力を持たない楽器の場合は、音響信号の出力から演奏音を特定することが必要である。単旋律演奏においては高い精度での音高推定の手法が確立されているものの、多重音からなるコード演奏においてはまだ十分な精度が実現されていない。そのため、本稿では対象をコード演奏のギターに限定して演奏音の特定を行う。

また、ギターに限らず自動譜めくりに共通する課題として、楽譜と異なる演奏への追従が挙げられる。人による実際の演奏には、弾き誤りや弾き飛ばし、テンポの揺らぎなど、演奏における不確定要素が含まれる。これらの要素に加えて、ギターの場合は演奏者の弦を押さえる場所に誤りがなくとも、力の入れ具合等の要因により、正しく発音することができない、という場面が存在する。そのため、ギターの場合は楽器からの出力から解析された演奏音が必ずしも演奏者の演奏意図と一致しているとは限らない。よって、ギターを対象とする自動譜めくりシステムにおいては、既存研究で自動譜めくりに要求されていた要素に加え、このようなギター特有の音の抜けや不完全な発音などを含む演奏への楽譜追跡も必須の要件である。

## 5. 提案手法

### 5.1 提案手法の概要

既存研究の多くは、演奏音を1音ずつ特定し、楽譜全体の中から演奏位置を特定するが、それゆえ音響信号が複雑になると演奏音の特定が困難になる、という課題が存在していた。そこで本稿では、ギターによるコード演奏の入力を、複数の音に分解して特定するのではなく、ひとつのコードとしてまとめてとらえる。これにより、音高や発音時刻などの演奏音の詳細な特定が不要になり、また演奏を単純化することができる。

図3に提案手法の概要を示す。本手法は、楽譜の処理と演奏のコード推定、そして楽譜追跡の3つの要素から構成される。まず、楽譜に記述されたリハーサルマークをもとに、楽譜を小区間に分割する。リハーサルマークとは、順に並んだ文字または数字を用いて楽譜上の位置を示す記号であり、複数人で楽譜を利用する際の基準となる[10]。また、本稿ではリハーサルマークで分割した区間のコード進行のことを「メロディ」と呼ぶ。そしてギターからの音響入力を一定時間で区切り、推定するコードの候補をメロディに含まれるものだけに限定してコード推定を行う。そして推定したコードとメロディで DP マッチングを行い、メロディの最後まで到達した時点で次のメロディへと遷移する。これを繰り返すことで、楽譜上での演奏位置を追跡する。

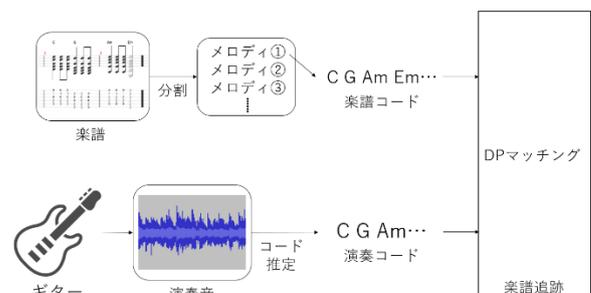


図3 提案手法の概要

## 5.2 本手法におけるコード推定

### 5.2.1 音楽におけるコード

音楽における「コード」とは音階の異なる音の組み合わせを表したものであり、構成音の中で最も音程が低く、コードの基準となる音を「ルート音」と呼ぶ。また、ルート音を基準に音階を相対的にあらわしたものを「度数」と呼び、ルート音からの音階の離れ具合を表している。例えばルート音が「ド」の音の場合、「ド」の音の度数は1度、「ソ」の音の度数は5度となる。コードは構成音の数に応じて、3和音、4和音、5和音などが存在し、その中でも一般に楽曲の中で頻りに用いられるものについては、度数の組み合わせで同じ響きを持つコードとして分類され、ルート音と合わせて「コードネーム」として表記される。コードネームの表記法や読み方については統一されていないため、本稿では日本で一般に用いられている方法で表記する。また、それぞれのコードにさらに最低音を加えたものを「分数コード（オンコード）」と呼び、例えば「Cメジャー」コードに最低音の「E」を加えた分数コードは「C/E」または「ConE」と表記される。コードには様々な種類が存在するため、本稿では分数コードは扱わず通常のコードとして処理を行い、コード推定ではギター楽曲で多く使用される18種類のコードを扱う。

### 5.2.2 本手法におけるコード推定の手法

本稿では関連研究で述べたコード推定の手法を用いて、類似度が最も高く、かつ閾値を超えたコードを推定コードとする。なお、本稿ではコードに含まれる構成音の存在比はすべて等しいものとし、類似度の閾値は0.6に設定している。

また、コード推定は一定区間の演奏に対して行うため、テンポが変動すると、推定区間とコードの切り替わるタイミングにずれが発生してしまい、正しく推定することができない。そのため、さらにテンポ推定を同時に行い、推定区間の調整をおこなうことでテンポの揺らぎによる影響を受けづらくする。なお、推定区間の初期値は楽曲のテンポを基準に拍数で設定する。例えばBPM120の楽曲を2拍ごとに推定する場合の推定区間は1秒となる。小節におけるコードの切り替わりが激しい場合、推定区間の拍数をコードの切り替わる間隔に合わせて設定する必要があるが、テンポが速い場合だと推定区間の時間が短くなってしまい、コード推定に十分な長さの音響信号を得ることができず、精度が低下してしまうことが考えられるため、推定区間の長さは楽曲に合わせて適切に設定する必要がある。

この手法によるコード推定の精度評価のため、以下の条件でコード推定を行った。

- ・ A から G#までの12音階で、メジャー、マイナー、セブンス、マイナーセブンス、サスフォーの5種類の合計60コードを演奏
- ・ それぞれのコードの演奏の長さ：1小節

- ・ テンポ：100
- ・ 推定候補とするコード：ギター楽曲で使用される場面の多いコード（12音階×18種）

表1にコード推定の結果を示す。それぞれの行が演奏したコードのルート音、列がそれぞれのコードの種類を表している。なお、正しく推定できたコードは60コード中32コードであり、正答率は53.3%であった。

表1 コード推定の結果

	M	m	7	m7	sus4
C	CM7(9)	Gsus4	C7	Cm7(9)	FM7(9)
C#	C#	C#m	C#7	C#m7	C#sus4
D	D	Dm	D7(9)	FM7(9)	A7sus4
D#	D#	D#m	D#7	D#m7	A#7
E	Eadd9	Em	E7	Em7	Esus4
F	FM7	Fm	F7	Fm7(9)	Fsus4
F#	F#M7	C#sus4	F#M7	F#m7(9)	F#sus4
G	GM7	Dsus4	GM7(9)	Gm7(9)	Gsus4
G#	G#M7	G#m	G#7	G#m7(9)	G#sus4
A	Aadd9	Esus4	A7(9)	Am7(9)	E7sus4
A#	A#	A#m	A#7(9)	A#m7	A#sus4
B	B	Bm	B7	Bm7	E7sus4

上記の手法ではコードの構成音の存在比をすべて等しいものとしているが、実際の演奏では和音を構成するそれぞれの音の大きさにはばらつきがあるため、正しく演奏されていても偶然に異なるコードの構成音との類似度が高くなってしまふことが考えられる。また、ギターで演奏されるコードの中には、特定のコードに1音を追加したような、構成音がほぼ同じものも存在する。これらの理由から、十分な精度のコード推定を行うことができなかったものと考えられる。

### 5.2.3 コード推定の精度改善

実験の結果から、上記の手法でのコード推定は精度が不十分であることが判明したため、コード推定の精度向上のための改善を行う。改善手法としては、すべてのコードを推定の候補とするのではなく、候補とするコードを追跡するメロディ内に含まれるコードのみに絞って推定を行う。これにより、誤推定の原因となる構成音の類似したコードが同じメロディ内に存在しない限りにおいてはそれらのコードの判別が不要になり、誤推定を減らすことができる。

### 5.2.4 コード推定精度改善の評価実験

ギターで演奏されるポピュラー音楽を対象に、実際のギター演奏を用いてコードの候補を絞った場合とそうでない場合でのコード推定の正答率の比較を以下のような条件で行った。

- ・ 使用した楽曲 / アーティスト名
  - 楽曲1：渚 / スピッツ
  - 楽曲2：夏色 / ゆず
  - 楽曲3：マリーゴールド / あいみょん
  - 楽曲4：ミライ / l'Arc~en~Ciel

- ・ テンポ：BPM120
  - ・ 演奏者：著者
  - ・ 演奏する楽譜の長さ：8小節
  - ・ コード推定の区間：2拍ごと
  - ・ コード推定の候補パターン
- ① それぞれの楽曲内に使用されているコード
  - ② ギター楽曲で使用される場面の多いコード(12音階×18種)
  - ③ 同じメロディ内のコード(楽曲4のみ)

実験の結果は以下になった。表2から表5はそれぞれの楽曲のコード推定結果であり、正答部分に色を付けている。また表6はそれぞれの楽曲とコード推定の候補パターンでの正答率を表したものである。

表2 楽曲1のコード推定の結果

正解	G	G	C	C	D	D	Em	Em	C	C	G	G	C	C	D	D
①	G	G	C	C	D	D	Em	Em	C	C	G	G	C	C	D	D
②	G	G	C	Cadd9	Dadd9	Dadd9	Em	Em	CM7	CM7	GM7	Gadd9	Cadd9	CM7	Dadd9	Dadd9

表3 楽曲2のコード推定の結果

正解	D#	Cm	G#	A#	D#	Cm	G#	A#	Gm	Gm	Cm	Cm	G#	G#m	D#	D#
①	D#	Cm	G#	A#	D#	Cm	G#	A#	Gm	Gm	Cm	Cm	G#	G#m	D#	D#
②	D#add9	G#M7	G#add9	A#M7	G#mM7	G#M7	G#add9	A#M7	D#M7	D#M7	G#M7	G#M7	G#	EM7	D#add9	D#M7

表4 楽曲3のコード推定の結果

正解	D	D	A	A	Bm	Bm	A	A	G	G	D	Bm	G	G	A	A
①	D	F#m7	A	A	Bm	Bm	A	A	G	Bm	D	Bm	G	G	A	A
②	DM7	Dadd9	AM7	Aadd9	Bm	Badd9	A7	Aadd9	GM7	GM7	DM7	Bm7	F	GM7	A	AM7

表5 楽曲4のコード推定の結果

正解	C	C	C	C	F	F	Fm	Fm	G#6	G#6	G#6	G#6	C	C	E7-9	G
①	C	C	C	Am7(9)	F	FM7(9)	F	FM7(9)	G#M7	G#M7	G#M7	Am7	C	Am7(9)	E7-9	G
②	Cadd9	Cadd9	CM7	CM7	FM7	FM7	F	Csus4	G#M7	G#M7	G#M7	Am7	Cadd9	Em7	EM7	Em7
③	C	C	C	C	F	F	F	Fm	G#6	G#6	G#6	C	C	E7-9	E7-9	G

表6 それぞれの楽曲と推定候補パターンの正答率

	楽曲1	楽曲2	楽曲3	楽曲4
①	1	0.9375	0.875	0.4375
②	0.3125	0.0625	0.0625	0
③				0.8125

すべての楽曲において、コードの候補を絞ったときは、絞らなかったときと比べて正答率が大きく向上している。また、楽曲4では曲中のコードに限定したことで正答率は向上したものの、まだ楽譜追跡に十分な精度は実現できていない。これは、楽曲内で使われているコードの種類が多く、またルート音が同じなど、類似したコードが多く使われていることが原因だと思われる。そのため、楽曲4のみさらにメロディ内のコードに候補を限定したところ、コード推定の精度が大きく向上した。このことから、コード推定において推定の候補となるコードの種類を限定することは、精度向上のために有効であるということが分かった。

### 5.2.5 コード推定結果の楽譜追跡への利用

実験結果から、改善手法によってコード推定の精度が向上することが示されたが、楽曲4の実験結果のように、推定候補にルート音が同じコードが多数含まれる場合、精度が大きく低下してしまう可能性がある。特に本稿では自動譜めくりを目的とするため、実際の演奏では演奏者の技術不足による演奏音の抜けなどが発生した場合に、演奏者の意図とは異なったコードになってしまうことがある。図4と図5はFコードとFM7コードそれぞれの押弦位置を表すコードダイアグラムであり、●が押弦位置を表し、図5のように押弦しない弦がある場合は左側の○×でその弦の音を鳴らすか否かを示す。例えば、図4と図5のように、FコードとFM7コードは構成音が類似しているため、Fコードを演奏しようとした場合に、押弦がうまくいかずにFM7コードになってしまうということが発生する。また、演奏が難しい場合に、類似したコードがその代用として使われることもある。Fコードは初心者にとっては難しいコードであるため、しばしばFコードの代用としてFM7コードが演奏される。

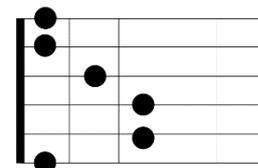


図4 Fコードの押弦位置

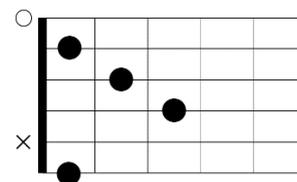


図5 FM7コードの押弦位置

これらの理由から、演奏音が必ずしも演奏者の演奏意図と同じであるとは限らないため、類似したコードを区別してしまうと、楽譜追跡において誤追跡の原因となることが予想される。そこで、楽譜追跡でコードの推定結果を扱う際には、メロディ内にルート音の同じコードが存在する場合、推定結果と楽譜との比較においてルート音のみを参照する。これにより、演奏者のミスや意図的なコードの代用によって楽譜とは異なるコードに推定されてしまっても、それらの区別を行わないため、影響を受けずに楽譜追跡を行うことができる。

### 5.3 楽譜追跡

コードの推定結果をもとに、ギター演奏をコードの進行としてとらえ単純化し、演奏コードと楽譜のコード名でDP

マッチングをすることで楽譜追跡を行う。演奏されたコード名を入力とし、比較する楽譜は実際の楽譜からコード進行のみを抽出したものを利用する。また、コード推定の候補と同様に追跡区間をメロディごとに分割し、メロディの最後まで演奏が行われたときに次のメロディへと遷移する。これにより、先行研究で発生していた、誤追跡によって発生する演奏位置の大幅なスキップや、楽譜が長くなることによる計算量の増加を防ぐ。

しかし一方で、メロディごとに分割すると、図6のようにメロディの切り替わるタイミングで弾き飛ばしが発生した場合、メロディの最後のコードに到達することができないため演奏中のメロディから次のメロディへの遷移ができず、追跡が止まってしまうという問題が発生する。本稿ではこのメロディ間の遷移をメロディ遷移と呼ぶ。特にメロディの切り替わるタイミングは楽曲の構造上の区切りでもあるため、G, Gsus4のような、ルート音が同じコードが連続するなどの特有のコード進行になることが多く、演奏ミスやコード推定での判別ができないなどの原因による弾き飛ばしが発生しやすい。



図6 メロディ遷移時に発生する弾き飛ばしの例

これを解決するため、演奏中のメロディだけを追跡範囲とするのではなく、次のメロディの先頭の3つのコードまでを追跡範囲に含める。これにより、メロディの切り替わるタイミングで引き飛ばしが発生した場合でもメロディ遷移が可能になる。また、初めからメロディの楽譜の最後に次のメロディの先頭3つのコードを追加しておき、演奏位置がメロディの終わりに近づいた時点で、推定候補のコードに追加したコードを加える。これにより、コードを追加したことによるコード推定への影響を抑える。また、提案手法では楽譜と演奏のマッチングのために演奏コードと楽譜のコード名を入力として Dannenberg らの手法を利用する。そのため、対象とする演奏で想定される、連続する間隔の長さに応じてマッチングを行う際の窓の大きさを適切に設定する必要がある。コード単位での楽譜追跡においては大きな弾き飛ばしは発生しにくいと考えられるため、本稿では弾き飛ばしの許容量を2コードまでとしている。そのため、3コード以上連続する誤りは発生しないことを前提とし、図7のように、次のメロディのコードを3コードまで参照することで、メロディの最後の2コードで間違いがあった場合でも、正しくメロディの遷移を行うことができる。

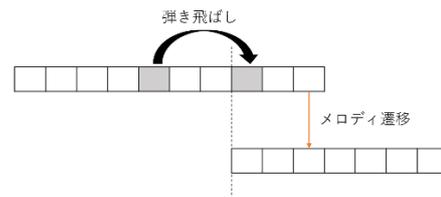


図7 メロディ遷移の改善

## 6. 自動譜めくりシステム

### 6.1 自動譜めくりシステムの構成

提案手法をもとに、自動譜めくりシステムを実装する。ギターからの音響信号を入力とし、演奏から楽譜追跡を行い、現在の演奏位置の楽譜を画面上に表示する。演奏の一時停止、再開にも対応しており、キー入力によって任意のメロディから演奏を開始することができる。また、指定フォルダに楽譜を追加することで任意の楽曲で利用することができる。

本稿では、Python のライブラリである“librosa”を用いてコード推定のためのクロマベクトルの算出とテンポ推定を行った。

### 6.2 楽譜の仕様

楽譜データは、表示する楽譜の JPEG ファイルと、楽譜の情報を記述した JSON ファイルを利用する。JPEG ファイルのファイル名は昇順での名前順がページの順番と一致するようにする必要がある。JSON ファイルが持つキーと対応する情報は以下の通りである。

- score : コード譜と対応するページ数による楽譜データ
- tempo : 初期値とする楽曲のテンポ
- page\_number : 総ページ数
- beat : コード推定を行う区間の拍数

“score”に記述する楽譜は、メロディごとの楽譜を要素とするリストで表現する。メロディの楽譜は図8に示すように、二次元のリストで1行目にコード、2行目にその次に演奏するコードの記載されているページ数を記述する。

メロディ①	Cadd9	D	Em	Bm7	Am7	D			
	1	1	1	2	2	2			
メロディ②	Cadd9	D	Em	Bm7	Am7	D	G	Gsus4	G
	2	2	2	2	2	3	3	3	3
メロディ③	Am7	D	G	D	Em	Cadd9			
	3	4	4	4	4	4			

図8 楽譜の形式

## 7. 評価

提案手法が弾き飛ばしや弾き誤り、テンポの揺らぎなど

の不確定要素を含む演奏にも頑健に追跡がおこなっているか評価するため、実際の楽曲とギター演奏を用いて、メロディ内での楽譜追跡、メロディ間での追跡、楽曲全体を通しての譜めくりの3つの観点からそれぞれ評価実験を行った。

### 7.1 メロディ内での楽譜追跡の評価

まず提案手法が弾き飛ばしや弾き誤り、テンポの揺らぎなどの演奏の不確定要素に対して効果的であるかを評価するために、メロディ内での遷移のみを対象に実験を行った。実験の条件は以下の通りである。

- ・ 使用した楽曲 / アーティスト名：憂, 燦々 / クリープハイブ
  - ・ テンポ：BPM110
  - ・ 演奏者：著者
  - ・ 演奏区間：メロディごとに分割したそれぞれ異なる演奏の5つの区間
  - ・ コード推定の区間：1拍ごと
  - ・ 演奏パターン
- ① テンポのゆらぎの有無
  - ② 弾き間違いの有無
  - ③ 弾き飛ばしの有無

これらの組み合わせの8組と、条件を意識しない自由な演奏の9パターン

以上の条件をもとに、実験を行った。

なお、テンポの揺らぎの有無については、テンポの揺らぎがない場合はメトロノームを聞きながら、極力一定のテンポで演奏を行うようにし、揺らぎがあるときは意図的にテンポを変動させながら演奏を行った。また、弾き間違いと弾き誤りについては、連続する間違いを2回までとし、各区間1~2回の演奏ミスを含むようにした。

表7 メロディ内での楽譜追跡の結果

①/②/③	無無無	有無無	無有無	無無有	有有無	有無有	無有有	有有有	自由
区間A	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	80.0%	60.0%	100.0%
区間B	90.0%	90.0%	90.0%	75.0%	90.0%	87.5%	87.5%	75.0%	66.7%
区間C	100.0%	92.9%	100.0%	83.3%	100.0%	83.3%	81.8%	75.0%	92.3%
区間D	100.0%	100.0%	100.0%	80.0%	83.3%	100.0%	80.0%	80.0%	100.0%
区間E	77.8%	100.0%	66.7%	87.5%	88.9%	71.4%	85.7%	71.4%	88.9%
平均	93.6%	96.6%	91.3%	85.2%	92.4%	88.5%	83.0%	72.3%	89.6%

表7は各演奏パターンの組み合わせで、それぞれの区間の演奏を行った際の楽譜追跡の正答率を表している。この結果から、テンポの揺らぎ、弾き誤り、弾き飛ばしがない場合と比べて、それらを含む演奏であっても、追跡精度を維持できていることがわかる。特にテンポの揺らぎの有無による影響が少なく、楽譜を暗記できていない段階での演奏はテンポの揺らぎが発生しやすいため、練習段階での利用には特に適しているといえる。また、全体として区間B、区間Eでの正答率が低下しているが、これはG、Gsus4、Gという進行がメロディの最後に存在し、コードの構成音

がGとGsus4では類似しているため、コード推定の段階で区別できていないことが原因であると考えられる。

### 7.2 メロディ遷移の評価

メロディ間の遷移部分に弾き飛ばし、弾き誤りを含む演奏に対する追従性能の評価のため、メロディ遷移時に次のメロディのコードを参照する場合とそうでない場合とでそれぞれで評価実験を行った。実験対象の楽曲は評価実験1と同様の条件にし、メロディにまたがる弾き誤り、弾き飛ばしがある場合とそうでない場合でそれぞれ10区間演奏し、そのうちのメロディ遷移の成功率で評価を行った。

表8 評価実験2の結果

弾き誤り	なし	あり
改善前	8/10	0/10
改善後	10/10	9/10

評価実験の結果は表8のようになり、演奏を行った10区間のうちの遷移の成功数を表し、改善前が次のメロディを参照しない場合、改善後が次のメロディを参照したときの結果である。この結果から、メロディ遷移部分のアルゴリズムの改善によって、正解率が大きく向上していることがわかる。特に、弾き誤りのない演奏においても、改善前はG、Gsus4、Gのようなルート音の変わらないコードが連続してメロディ遷移の直前にある場合に追跡が失敗していたが、改善によって正しく追跡ができるようになっていく。また、弾き誤りがある演奏において改善前は全く追跡を行うことができなかったのに対して、改善後では高い精度で追跡が可能になった。弾き誤りがある演奏では改善後でもメロディ遷移が失敗する場面があったが、これは次のメロディのコードに、現在のメロディのコードと構成音が類似したコードが含まれており、コード推定の際に正しく推定ができなかったことが原因であると思われる。構成音が類似したコードとの推定誤りが発生する問題については、ルート音が一致しているもの同士であれば提案手法で解決ができているが、ルート音が一致していない場合でも構成音が類似しているコードの組み合わせは存在し、提案手法だけでは解決できていないため、この問題の改善は今後の課題である。

### 7.3 自動譜めくりシステムの評価

提案手法の評価のために、実装した自動譜めくりシステムを用いて、ギターによって演奏される、日本のポピュラー楽曲3曲をそれぞれ2回演奏し、譜めくりの成功回数を計測した。システム利用時に想定される演奏と同様に、演奏者は楽譜通りの演奏を意識して演奏を行い、一部弾き誤りやテンポの揺らぎが含まれている。また、譜めくりが失敗し、追跡が止まってしまった場合には、一度演奏を停止し、次のページから追跡を再開している。

表 9 自動譜めくりシステムの評価実験の結果

	楽曲1	楽曲2	楽曲3
譜めくり回数	13	26	14
平均成功回数	12.5	23.5	12
成功率	96.2%	90.4%	85.7%

表 9 は実験の結果であり、3 つの楽曲すべてにおいて 9 割近い成功率で譜めくりを行うことができている。しかし一方で、譜めくりの直前に誤追跡の発生しやすい演奏が存在する、同じメロディ内に「CM7」と「Am7」のような構成音の類似したコードが存在する、などの場合には、譜めくりが適切に行えないことがあったため、これらの改善による譜めくりの成功率向上が今後の課題である。

## 8. 考察

本稿では、音響信号を入力とした自動譜めくりにおける様々な問題に対し、ギターにおける譜めくりの特性に着目して解決すべき課題を明確にすることにより、音高推定や楽譜追跡などのそれぞれの構成技術の精度が不十分であっても、表 9 で示したようにシステム全体としての譜めくりの精度を 9 割近くまで向上させることができた。

また本稿では音楽の演奏という自由度の高いものを対象としているため、システムの仕様としてどの程度の誤りを含む演奏まで確実に追従可能であるかを明確に定義することが難しい。そのため、テストに利用する演奏パターンを決めることやその再現も難しく、現状では十分なテストを行えていない可能性がある。また、テストを実施する人の技量や使用するギターの種類の違いなどの環境による様々な影響も想定されるため、テストの担当者の確保や詳細な条件の設定が難しい。そのため、システムの評価の際にはこのような点を考慮して評価方法を考える必要がある。

## 9. おわりに

本稿では、入力となるギター演奏のコード推定を行い、メロディごとに楽譜を分割したうえで DP マッチングによる楽譜追跡を行うことで、複雑な多重音の演奏であるギター演奏を単純化し、ギター演奏を対象とした自動譜めくりシステムのための楽譜追跡手法の提案を行った。

また、提案手法の評価のために、提案手法を用いた自動譜めくりシステムを実装し、実際の演奏で譜めくりを行ったところ、演奏者が故意に演奏の誤りを発生させない場合においては、誤りを含む演奏においても楽曲を通して高い成功率で正しく譜めくりを行うことができる事が示された。

今後の課題としては、単旋律とコードの両方を含んだ演奏への対応が挙げられる。実際のギター演奏では、コード演奏を主に行うバックギンギンにおいても単音の演奏が含まれ

ることもある。単旋律演奏の楽譜追跡についてはすでに高い精度の手法が確立されているため、このような演奏において、単旋律演奏とコード演奏の判別を行うことができれば、本手法と組み合わせることで単旋律とコードの両方を含んだ演奏に対しても追跡を行うことができると考えている。

本稿では、自動譜めくりの実現に関して、その対象をギターのコード演奏に絞り、楽曲や演奏の特徴を分析して要件を明確にすることにより、個別の構成技術の精度が不十分でも全体として譜めくりの精度を向上させることができた。また、自動譜めくりのような扱う対象の自由度が高いシステムの開発を行う際には、様々な要因による影響が考えられるため、仕様を明確にすることや、十分なテストを行うことが難しいため、これらの点を考慮して評価方法を考えることが必要である。

## 参考文献

- [1] Shibata Kentaro, Nakamura Eita, Yoshii Kazuyoshi, "Non-local musical statistics as guides for audio-to-score piano transcription", *Information Sciences*, vol.566, p.262-280, 2021.
- [2] 田部井 泰彦, "ウェーブレット変換を用いた単一音信号の音高推定法", *大学院研究年報 理工学研究科編*, no.36, 2006.
- [3] 安部 翔, 小田 弘良, 松島 俊明, "自動譜めくりシステムのための多重音に対する音高・音源数的高速推定法", *第 74 回全国大会講演論文集*, vol.2012, no.1, p.389-390, 2012.
- [4] 小倉 淳, 松島 俊明, "楽譜の自動譜めくりシステムのための演奏位置推定方法", *第 73 回全国大会講演論文集*, vol.2011, no.1, p.265-266, 2011.
- [5] 中村 栄太, 武田 晴登, 山本 龍一, 齋藤 康之, 酒向 慎司, 嵯峨山 茂樹, "任意箇所への弾き直し・弾き飛ばしを含む演奏に追従可能な楽譜追跡と自動伴奏", *情報処理学会論文誌*, vol.54, no.4, p.1338-1349, 2013.
- [6] 三浦 雄文, 赤羽 歩, 佐藤 誠, 津田 貴生, 井上 誠喜, 小宮 山 撰, "音響信号の振幅を用いるオーケストラ楽曲の楽譜追跡", *映像情報メディア学会誌*, vol.61, no.7, p.997-1005, 2007.
- [7] 酒向 慎司, "音響信号によるリアルタイム楽譜追跡と能動的演奏支援システムに関する研究", *電気通信普及財団 研究調査助成報告書*, no.31, 2016.
- [8] 松岡 保静, 渡部 瑞季, "from NTT DOCOMO テクニカル・ジャーナル 音楽のコード認識技術とその応用", *NTT 技術ジャーナル*, vol.29, no.9, p.30-37, 2017.
- [9] DANNENBERG R., "An On-Line Algorithm for Real-Time Accompaniment", *Proceedings of the International Computer Music Conference 1984*, p.193-198, 1984.
- [10] Steinberg Documentation, "リハーサルマーク", [https://steinberg.help/dorico\\_elements/v2/ja/dorico/topics/notation\\_reference/notation\\_reference\\_rehearsal\\_marks\\_c.html](https://steinberg.help/dorico_elements/v2/ja/dorico/topics/notation_reference/notation_reference_rehearsal_marks_c.html), (参照 2021/01/20)