

## オンセット検出を用いたギターコード演奏楽音の認識に関する一検討\*

○近藤駿, 鴨澤秀郁, 田中元志 (秋田大)

## 1 はじめに

音楽を扱うサービスが広く普及してきており、音楽情報処理に関する研究の重要性が高まっている [1]。その中でも、楽曲の分析や保存のために、演奏楽音から自動で楽譜を作成する自動採譜技術が望まれている。楽音からの採譜は、初めに音高認識を行う必要があるが、単音の場合では、基本周波数を検出することによって音高を認識することができる。多重音の場合は、発音音高の倍音の周波数が、他の発音音高およびその倍音の周波数に近い値となるため、単純な周波数検出による認識は難しい。コードの種類が分かれば、コード譜を作成することができ、採譜の補助になる。コード認識の方法として、クロマベクトルを用いた手法について多く検討されている [2, 3]。

筆者らは、コードの構成音をデータベース (DB) とし、そこから求まる倍音の情報を含む構成音情報を用いたコード認識方法を提案した [4]。理想的なコード演奏楽音を 800 音以上用意し認識を行った結果、全体で 94% 以上の認識率が得られ、その有効性を示した。オンセット (発音) 検出とこの手法を用いて、実際に連続して演奏された楽音のコード認識を試みたが、よい認識率は得られなかった [5]。

本研究では、構成音情報を用いたコード認識方法 [4] を用いて、オンセット検出後の数フレームについてコードを推定し、それらの結果を多数決処理することによって、演奏されたコードを順次認識する方法を検討する。テンポと音符長を変えて、複数の異なるコードを連続して演奏した楽音を用いて認識実験を行い、認識率を評価する。

## 2 演奏楽音のコード認識方法

ギターコード演奏楽音を、フレーム長 100 ms (ハニング窓)、シフト長 5 ms、 $2^{16}$  点 FFT で時間-周波数解析し、以下の手順でコードを認識する。

- (1) スペクトルフラックスを用いたオンセット検出
- (2) オンセット検出時刻から 5 フレームを抽出
- (3) フレーム毎にコード認識
- (4) 結果を多数決処理して発音コードを決定
- (5) 次のオンセットを検出し、(2)~(5) を繰り返す

これまで切り出したすべてのフレームについてコードの識別を行っていたが、本手法により、より少ない計算量でのコード認識が期待される。

## 2.1 オンセットの検出方法

オンセット検出に、次式で求められるスペクトルフラックス (SF; Spectral flux) を用いた。

$$SF[n] = \sum_k \max(0, (X[n+1, k] - X[n, k])^2) \quad (1)$$

ここで、 $k$  はパワースペクトル  $X$  における周波数ビン、 $\max(A, B)$  は  $A$  と  $B$  のうち大きい方を返す関数である。しきい値を超えた SF のピークを検出し、その時刻をオンセットとして検出する。

ギターコード演奏楽音 (C コードを 8 回演奏) のオンセット検出結果例を Fig. 1 に示す。(a) は時間波形、(b) はそのスペクトログラム、(c) は SF とオンセット検出結果である。図 (c) 中の点線は、最も精度よくオンセット検出できるように決めたしきい値である。発音に対応したオンセットを検出できていることが確認できる。なお、今回の解析条件では、実際の発音時間との誤差は約 27 ms であった。この誤差を考慮すると、4 分音符の演奏でテンポ 380 程度までの楽音が認識可能と考えられる。

## 2.2 コードの認識方法 [4]

コード認識に用いる、構成音情報を用いた認識アルゴリズムについて、以下にまとめる。

- (1) 解析フレームのスペクトルから、複数の倍音に対応するピークをもつ音高を抽出し、その音高に対応する MIDI ノートナンバーに変換する。
- (2) 最も低い音階番号  $n_L$  から  $n_L + 31$  までの成分を演奏音情報として抽出する。
- (3) 抽出されたノートナンバーを最も低いノートナンバー  $n_L$  を 1 として正規化し、演奏音情報 PSI (Played Sound Information) とする。
- (4) DB (コード構成音のみ) から倍音情報を含めた構成音情報 CTI (Constituent Tone Information) を求める。PSI と CTI を比較し、一致度が最も大きい“コード候補”を選択する。
- (5) コード候補の DB の根音に対応する音階を検出し、解析フレームのコードを推定する。

## 3 ギターコード演奏楽音の認識

## 3.1 実験用コード演奏楽音

コード演奏楽音を、エレキギターの音がサンプリングされた VSTi 音源 [6] を用いて、DAW (Digital

\* A study on chord recognition of musical guitar sound using onset detection, by KONDO, Shun, KAMOZAWA, Hidefumi and TANAKA, Motoshi (Akita University).

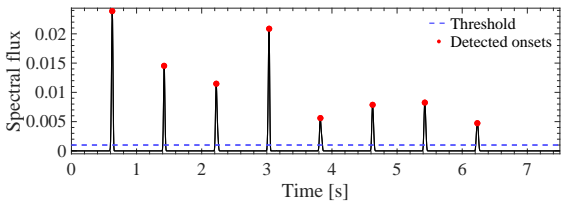
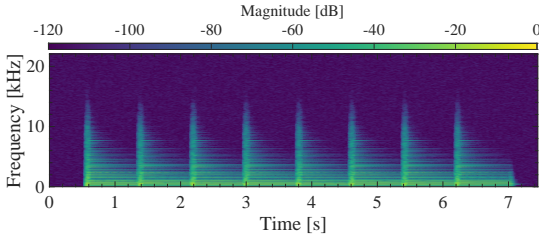
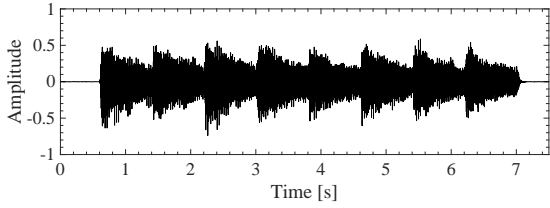


Fig. 1 Example result of time-frequency analysis and onset detection of musical guitar sound.



Fig. 2 Musical scores for playing guitar chords.

Audio Workstation) 上で作成した。楽音は、すべてダウンストロークを模擬し、Fig. 2 に示す楽譜 (1 小節分) の通り演奏された。Pattern 1 は、4 分音符 8 回のストロークとし、2 小節の楽音である。2 小節とも同じコード (C,  $C_{M7}$ ) を演奏した場合、1 小節でコードを変えた場合 ( $C \Rightarrow G$ ,  $C \Rightarrow C_m$ ) とし、テンポによる影響 (演奏音の時間間隔の違い) を評価する。Pattern 2 では、別の VSTi 音源 [7] を用いて、コードを 1 小節毎に  $C \Rightarrow F \Rightarrow G \Rightarrow C$  と変化させた 4 小節の楽音を用意した。各演奏パターンで、表 1 に示すようにテンポ (BPM) を変えた楽音を用意した。テンポ 300 および 360 は、テンポ 150 および 180 でそれぞれ 8 分音符を演奏した場合に相当する。

### 3.2 コード認識結果と考察

コード認識を含めた結果 (認識率) を表 2 にまとめる。Pattern 1 では、テンポ 180 までのコードはほぼ 100% 認識できたが、演奏テンポが速い場合に誤検出が増えた。また、Pattern 2 でも、音符長が短い場

Table 1 Tempo of the musical sound.

Pattern	Tempo (BPM)
Pattern 1	75, 120, 150, 180, 300, 360
Pattern 2	120, 150

Table 2 Result of the chord recognition.

Pattern	Recognition rate [%]
Pattern 1	76.3%
Pattern 2	78.1%
Overall	77.2%

合に誤認識が多い結果となった。全体としての認識率は 77.2% であった。オンセット検出については、全体で 140 回の発音に対して、誤検出が 6 回であった。Pattern 1 のテンポ 300 と 360 の場合に誤検出が多く、SF の時間変化を確認すると、発音に対するピークを検出できない箇所があった。発音間隔が短い場合、周波数成分が減衰しないうちに次の発音に移行するため、同じコードの場合、フレーム間の周波数スペクトルの差が小さいためと考えられる。オンセット検出の高精度化が必要と考えられる。

## 4 おわりに

オンセット検出後の数フレームからコードを多数決処理し、演奏されたコードを順次推定する方法を検討した。オンセットについては比較的高い精度で検出できたが、テンポが速い場合のコード認識率は思わしくなかった。今後は、演奏楽音を増やし、コード認識の高精度化、機械学習の利用などについて検討する。

## 参考文献

- [1] 亀岡他: “音楽音響信号処理技術の最先端”, 信学誌, 98(6), 467-474.
- [2] T. Fujishima: *1999 Int. Computer Music Conf.* (Beijing, China), 464-467, 1999.
- [3] E. Benetos, et al.: *IEEE Signal Processing Magazine*, **36**(1), 20-30, 2019.
- [4] Y. Bando, et al.: *IEEJ Trans. Electr. Electron. Eng.*, **17**(1), 103-109, 2022.
- [5] 近藤駿他: 音学音楽音響研資, MA2022-63, 2022.
- [6] Applied Acoustics Systems Inc.: “Strum Session 2”, <https://www.minet.jp/brand/aas/strum-session/> (閲覧日 2023.11.2)
- [7] “Standard Guitar”, <https://unreal-instruments.wixsite.com/unreal-instruments/standard-guitar/> (閲覧日 2023.11.2)